$$(S_{I,J}f)(x,y)=\sum_{i=0}^{2^I}\sum_{j=0}^{2^J} z_{i,j}\Phi\left(2^I\frac{x-a}{b-a}-i\right)\Phi\left(2^J\frac{y-c}{d-c}-j\right) \qquad (13)$$

We will represent the matrix of $z_{i,j}$ as the sequence of input approximation coefficients $\{c^0\}$. Process 68 concludes by storing the sequence $\{c^0\}$ in image memory 67.

It has been found that the sequence $(S_{I,J}f)(x,y)$ has important properties that are useful in the compression of documents according to the preferred embodiment of the invention, and particularly in the compression of two-color, or binary, documents. Firstly, the sequence $(S_{I,J}f)(x,y)$ is a bilinear spline function that is continuous over the rectangle [a, b] by [c, d], and its restrictions on the subdivided sub-rectangles $[x_i, x_{i+1}]$ by $[y_j, y_{j+1}]$ for integers i from 0 to 2^I, and integers j from 0 to 2^J are all bilinear polynomials. Secondly, the sequence $(S_{I,J}f)(x,y)$ agrees exactly with the sampled input signal sequence $f(x_i, y_j)$ at the grid points. Thirdly, the sequence $(S_{I,J}f)(x,y)$ has the second order of approximation, in that:

$$|(S_{I,J}f)(x,y)-f(x,y)|\le(constant)\,2^{-2(I+J)} \qquad (14)$$

for all points (x,y) within the rectangle [a, b] by [c, d], and for all smooth functions f(x,y), with the constant being independent of the number of grid points and grid lines.

Following projection process 68, spline-wavelet decomposition may now be done, in the general sense, by decomposition of the input approximation coefficient

sequence $\{c^0\}$ into low-pass and band-pass components. The ability to perform such decomposition is due to the multi-resolution analysis properties discussed above.

For purposes of explanation, in a one-dimensional case, let one-dimensional functions $f_j(x)$ and $g_j(x)$ be represented as follows:

$$f_j(x) = \sum_k c_k^j \phi(2^j x - k) \tag{14}$$

$$g_j(x) = \sum_k d_k^j \psi(2^j x - k) \tag{15}$$

From multi-resolution analysis, an approximated signal $f_m(x)$ can be decomposed into the sequence:

$$f_m(x) = g_{m-1}(x) + g_{m-2}(x) + \ldots + g_{m-m'}(x) + f_{m-m'}(x) \tag{16}$$

where $m' < m$, and where $g_j(x) \in W_m$ and $f_j(x) \in V_j$.

Recalling from multi-resolution analysis that spline space $V_{j+1} = V_j \oplus W_j$, that $V_j \subset V_{j+1}$ and that $W_j \subset W_{j+1}$, in terms of the functions $f(x)$ and $g(x)$, decomposition may be considered as:

$$f_{j+1}(x) = f_j(x) + g_j(x) \tag{17}$$

In this embodiment of the invention, the function $g_j(x)$ provides the higher-frequency details of the image (i.e., edges and corners) at the j^{th} level of decomposition, while the function $f_j(x)$ provides a low-frequency

approximation of the image. While one-dimensional functions f(x) and g(x) are used hereinabove for explanation, it is of course to be understood that two-dimensional decomposition functions f(x,y) and g(x,y), as useful in the decomposition of a two-dimensional functions representative of a document, may be obtained in a similar way (i.e., via the tensor product).

One can thus determine a relationship between a higher-level approximation coefficient sequence $\{c^{j+1}\}$ and its lower-level component approximation coefficient sequences $\{c^j\}$, $\{d^j\}$ as follows:

$$c_k^j=\sum_l a_{2k-l}c_l^{j+1} \tag{18}$$

$$d_k^j=\sum_l b_{2k-l}c_l^{j+1} \tag{19}$$

indicating that spline-wavelet decomposition according to the present embodiment of the invention may be performed by convolution of the approximation coefficient sequences with decomposition sequences $\{a_n\}$, $\{b_n\}$.

Accordingly, referring now to the illustration of the generalized method of Figure 7 for the decomposition of a document, subprocesses 70 and 71 of decomposition 48 are performed upon the input signal coefficient sequence $\{c^0\}$; Figure 7 shows subprocesses 70, 71 operating upon a generalized sequence $\{c^j\}$, considering that the input to processes 70 and 71 will be a previously decomposed component sequence in the case where decision 51 (Figure 2) requires additional decomposition to obtain the

desired compression ratio. According to this embodiment of the invention, processes 70, 71 analyze the input sequence in row-wise fashion first. Subprocess 70 performs the convolution of the sequence $\{c^j\}$ with a coefficient sequence $\{a_n\}$ that corresponds with the selected linear B-spline function, as noted above relative to Figure 5a. This convolution of process 70 is followed by a downsampling operation in which every other discrete result of the convolution operation is discarded, for example each odd-numbered sample; as will be evident from the detailed description hereinbelow, this downsampling may be an inherent result of the convolution operation. The result of process 72 is sequence $\{c^{j-1}\}$, which is then stored back into image memory 67.

Similarly, to derive the row-wise high-pass component of the input signal, input signal coefficient sequence $\{c^0\}$, or generally the sequence $\{c^j\}$ (whether initiated by projection process 68 or repetitively from a prior decomposition) is convolved with coefficient sequence $\{b_n\}$ that corresponds with the selected wavelet function in process 71. This convolution of process 71 is followed by, or incorporates, a downsampling operation in which every other discrete result of the convolution operation is discarded, for example each odd-numbered sample. The result of process 73 is the sequence $\{d^{j-1}\}$, which is then stored back into image memory 67 along with the low-pass sequence $\{c^{j-1}\}$ in process 72.

Subprocesses 73, 74 are then performed, in which the decomposition of the results $\{c^{j-1}\}$, $\{d^{j-1}\}$ are similarly decomposed, but in a column-wise fashion. As such, subprocess 73 convolves the blur image, i.e. $\{c^{j-1}\}$, with both spline function coefficient sequence $\{a_n\}$ and wavelet function coefficient sequence $\{b_n\}$, followed by or

incorporating therein a downsampling by a factor of two. The result of subprocess 73 is the blur portion of the two-dimensional image (referred to as the LL component) corresponding to the column-wise low-pass filtered component of row-wise low-pass filtered result $\{c^{j-1}\}$, and a higher-frequency component (referred to as the LH component) corresponding to the column-wise low-pass filtered component of the row-wise high-pass filtered result $\{d^{j-1}\}$. Similarly, subprocess 74 convolves and downsamples the wavelet component $\{d^{j-1}\}$ with both spline and wavelet function coefficient sequences $\{a_n\}$, $\{b_n\}$, respectively. The result of subprocess 74 is the high-frequency component of the two-dimensional image referred to as the HL component, which corresponds to the column-wise high-pass filtered component of the row-wise low-pass filtered result $\{c^{j-1}\}$, and the highest frequency component referred to as the HH component, which corresponds to the column-wise high-pass filtered component of the row-wise high-pass result $\{d^{j-1}\}$. The component results LL, LH, HL, HH are stored back into image memory 67 in process 75, for quantization by process 50 of the flow of Figure 2.

(i) Dual Base Wavelet Decomposition

According to a first embodiment of the invention, decomposition process 48a takes advantage of the duality principle for certain wavelets, as described in Chui, et al., "Fast integral wavelet transform on a dense set of time-scale domain", CAT Report 310 (Center for Approximation Theory, Texas A&M University, September 1993). Through use of the duality principle, it has been discovered that each element of both decomposition coefficient sequences $\{a_n\}$, $\{b_n\}$, and of both reconstruction coefficient sequences $\{p_n\}$, $\{q_n\}$ may be expressed as a rational number. Integer scaling of these

sequences thus allows for all of the matrix operations of process 48a to be integer operations, and can thus be performed very quickly and efficiently by integer-based DSP 60, such as the TMS 320C25 or the like, in spline-wavelet compressor 20. It is therefore preferred, according to the present invention, that the default condition of spline-wavelet compressor be set to perform decomposition according to process 48a, using the dual wavelet approach.

For purposes of explanation, and as noted in the above-referenced Chui et al. paper, a "dual" function $\tilde{\phi}_m(x)$ of the mth order B-spline function $\phi_m(x)$ exists, such that:

$$\tilde{\Phi}_m(x) \in clos_{L^2} \quad span\{\phi_m(x-k) \mid k \in Z\} \tag{20}$$

and such that:

$$\int \phi_m(x-k)\tilde{\Phi}_m(x-l)\,dx = \delta_{k,l} = \begin{cases} 1 & k=l \\ 0 & k \neq l \end{cases} \tag{21}$$

It is preferred that the order m be even-numbered; the most useful case is the linear case (m=2), in which:

$$\Phi_2(x) = \frac{1}{6}(\tilde{\Phi}_m(x+1) + 4\tilde{\Phi}_2(x) + \tilde{\Phi}_2(x-1)) \tag{22}$$

Because of this relationship, one may perform wavelet decomposition in dual space, using linear dual spline function $\tilde{\phi}_2(x)$ together with linear dual wavelet function $\tilde{\psi}_2(x)$ (i.e., the dual of linear wavelet function $\psi_2(x)$).

Linear wavelet function $\psi_2(x)$, in this example, is preferably the following function:

$$\psi_2(x) := \sum_n q_n N_2(2x-n) \tag{23}$$

where, for m generally:

$$q_n = \frac{(-1)^n}{2^{m-1}} \sum_{l=0}^{m} \binom{m}{l} N_{2m}(n+1-l), \quad m=0,1,\ldots,3m-2 \tag{24}$$

For a discussion of these functions in the general case (for order m), attention is directed to Chui, <u>An Introduction to Wavelets</u> (Academic Press, 1992), incorporated herein by this reference; particular attention is directed to pages 182-186 in this regard. By way of the duality principle noted above, the dual ot linear wavelet function $\psi_2(x)$ may therefore be expressed as follows:

$$\tilde{\psi}_2(x) = -\frac{1}{2} \sum_{k=-\infty}^{\infty} c_k^4 \psi_2(x+3-k) \tag{25}$$

<u>Id</u>., at 185.

By way of further explanation, use of the dual base wavelet decomposition technique will first be described relative to a one-dimensional sampled input signal $f(k/2^J)$, where 2^J is the number of sample points in the

input sequence and where k is the sample point index in the sequence. Projection of the sampled input signal onto spline space results in the function $f_J(x)$ as follows:

$$f_J(x)=\sum_k f(\frac{k}{2^J})\phi_2(2^J x+1-k)$$

(26)

The function $f_J(x)$ thus provides a continuous representation of the sampled input signal $f(k/2^J)$. As noted above relative to process 68 of Figure 7, since the approximation coefficient sequence $\{c_k^0\}$ is exactly the original sequence of input signal samples, projection of the function $f_J(x)$ into spline space as a set of approximation coefficients requires no computation in this case and results in the original sampled input signal sequence $f(k/2^J)$.

Next, using the duality principle according to this embodiment of the invention, the continuous function $f_J(x)$, as represented by the approximation coefficient sequence represented directly by the sampled input signal sequence $f(k/2^J)$, is changed into a dual approximation coefficient sequence by way of a coefficient sequence c_k^J determined by equation (22) above as follows:

$$c_k^J=\frac{1}{6}\{f(\frac{k}{2^J})+4f(\frac{k+1}{2^J})+f(\frac{k+2}{2^J})\}$$

(27)

Referring back to the decomposition tree of Figure 5a, the following dual space decomposition formulae may be readily derived:

$$c_k^j = \sum_{l=2k-m}^{2k} a_{2k-l} c_l^{j+1} \tag{28}$$

$$d_k^j = \sum_{l=2k-3m+2}^{2k} b_{2k-l} c_l^{j+1} \tag{29}$$

using the finite decomposition sequences a_k, b_k, as follows:

$$a_k = 2^{-m}\binom{m}{k} \qquad k=0,1,\ldots,m \tag{30}$$

$$b_k = \frac{(-1)^k}{2^m}\sum_{l=0}^{m}\binom{m}{l}\Phi_{2m}(k+1-l) \qquad k=0,\ldots,3m-2 \tag{31}$$

In the linear (m=2) case, the sequences a_k, b_k equal:

$$\{a_0, a_1, a_2\} = \frac{1}{4}\{1,2,1\}, \quad a_k=0 \textit{ for all other } k \tag{32}$$

$$\{b_0, b_1, b_2, b_3, b_4\} = \frac{1}{24}\{1,-6,10,-6,1\}, \quad b_k=0 \textit{ for all other } k \tag{33}$$

Further, the reconstruction sequences $\{p_k\}$, $\{q_k\}$ equal the decomposition sequences $\{a_k, b_k\}$. In addition, reference to the summation limits in equations (28), (29) will show that the convolution operation will also result in

downsampling of the results, by a factor of two. As will be indicated hereinbelow, the results of this convolution operation will provide both low- pass and high-pass component results that can be stored in the same memory space as that which originally contained the input sequence.

As is evident from the foregoing, the decomposition sequences $\{a_k, b_k\}$ will be quite advantageous from the standpoint of implementation into modern data processing equipment, since each sequence is a finite sequence of rational numbers. The decomposition routine is thus very similar to a digital FIR (finite impulse response) filter. As such, dual base wavelet decomposition process 48a is very efficient when performed by spline-wavelet compressor 20 on two-dimensional input data representative of a document, as will now be described relative to Figures 8 and 9.

Decompression process 48a of Figure 8 begins with process 76, in which the two-dimensional input signal coefficient sequence $\{c^j\}$ is projected into dual spline space. In this example, the j term in the nomenclature $\{c^j\}$ indicates the order of decomposition, such that the coefficient sequence $\{c^0\}$ refers to the original input signal coefficient sequence. In the two-dimensional case, input signal coefficient sequence $\{c^0\}$ refers to the sequence $(S_{I,J}f)(x,y)$. Process 76 generates the dual representation $\{\tilde{c}^j\}$ as a series of coefficients according to the following sequence, analogously to equation (22) above, as follows:

$$\tilde{c}_{i,j}=(c_{i-1,j-1}+c_{i-1,j+1}+c_{i+1,j_i}+c_{i+1,j+1})+16c_{i,j}+4(c_{i-1,j}+c_{i,j-1}+c_{i,j+1}+ \qquad (34)$$

where

$$c_{-1,j} = c_{2^I+1,j} = c_{i,-1} = c_{i,2^J+1} = 0 \quad \textit{for all } i,j$$

As a result, one may express the relationship between the spline space coefficient sequence $\{c^j\}$ and its dual coefficient sequence $\{c^j\}$ as follows:

$$(S_{I,J}f)(x,y) = \{c^j\} = \frac{1}{36}\sum_{i=0}^{2^I}\sum_{j=0}^{2^J} \tilde{c}_{i,j}\tilde{\Phi}_2\left(2^I\frac{x-a}{b-1}-i\right)\tilde{\Phi}_2\left(2^J\frac{y-c}{d-c}-j\right) \tag{35}$$

Upon completion of process 76, the dual coefficient sequence $\{c^j\}$ is stored back into image memory 67.

Process 78 and 80 are then performed by way of which the row-wise decomposition processes are performed by DSP 60 using integer operations, as enabled by the dual base wavelet technique according to this embodiment of the invention. As noted above, the input sequence $\{c^j\}$ consisted of a two-dimensional array of values in PGM format; as such, the dual sequence $\{c^j\}$ similarly consists of such an array. Process 78 performs convolution of each row of the dual array sequence $\{c^j\}$ with a sequence $\{a_k\}$ derived as noted above, in the manner of a low-pass filter. In this example, where order m = 2 (i.e., linear spline-wavelet analysis), the convolution of process 78 consists of the point-wise multiplication of each row in dual sequence $\{c^j\}$ with a sequence $\{a_k\}$ derived according to equation (28) noted above:

$$\tilde{c}_k^{j-1} = \sum_{l=2k-m}^{2k} a_{2k-l} \tilde{c}_l^j \tag{28}$$

where k is the position of each entry within the row of dual array sequence $\{c^j\}$. As noted in equation (32) for the linear case, the sequence $\{a_k\}$ is defined as:

$$\{a_0, a_1, a_2\} = \frac{1}{4}\{1,2,1\}, \quad a_k = 0 \text{ for all other } k \tag{32}$$

The 1/4 term may be applied to the resultant data later, allowing the convolution of process 78 to be performed solely with integer values. As noted above, the convolution of process 78 results in a convolution result $\{c^{j-1}\}$ that is inherently downsampled by a factor of two. For example, only the even-numbered values of the convolution results may be retained (given the summation limits noted above), with the odd-numbered values discarded. As will be evident hereinbelow, this downsampling allows for the same image memory space to be used to maintain the decomposed image.

Similarly, process 80 performs convolution of the dual space array sequence $\{c^j\}$ with a sequence $\{b_k\}$ derived as noted above, in the manner of a high-pass filter. As in the case of process 78, where the order m = 2 (i.e., linear spline-wavelet analysis), the convolution of process 80 consists of the point-wise multiplication of each row in dual sequence $\{c^j\}$ with a sequence $\{b_k\}$ according to equation (29) noted above:

$$\tilde{d}_k^{j-1} = \sum_{l=2k-3m+2}^{2k} b_{2k-l}\tilde{c}_l^j$$

(29)

As noted in equation (33) above, for the linear case, the sequence $\{b_k\}$ may be calculated as follows:

$$\{b_0, b_1, b_2, b_3, b_4\} = \frac{1}{24}\{1, -6, 10, -6, 1\}, \quad b_k = 0 \text{ for all other } k$$

(33)

Again, the 1/24 division may be performed at a later time, to allow the convolution of process 80 to be performed using only integer values. As noted above, the row-wise convolution of process 80 provides results $\{d^{j-1}\}$ that are downsampled by a factor of two, e.g. by retaining only the even-numbered entries.

Upon completion of the row-wise decomposition and downsampling of processes 78, 80, the downsampled decomposed sequences $\{c^{j-1}\}$, $\{d^{j-1}\}$ are stored in image memory 67, in a form in which the left-hand side of the image corresponds to the low frequency results $\{c^{j-1}\}$, and in which the right-hand side corresponds to the high frequency results $\{d^{j-1}\}$. Attention is directed to Figure 9 for a visual representation of the operations of processes 78, 80. Array 90 indicates the contents of input dual array sequence $\{c^j\}$, prior to row-wise decomposition. The inherent downsampling in the results of each of processes 78, 80 provides a combined downsampled sequences $\{c^{j-1}\}$, $\{d^{j-1}\}$ that is stored by process 82 in image memory 67 in the same memory space as the original array, as shown by array 92 of Figure 9.

The results $\{c^{j-1}\}$, $\{d^{j-1}\}$ of the row-wise decomposition are then subjected to column-wise decomposition, for both low-pass and high-pass cases, in processes 84, 86 respectively. Process 84 retrieves each column of the stored results for sequence $\{c^{j-1}\}$ (from array 92), and performs a decomposition into another pair of low and high frequency sequences in the same manner as process 78 discussed above, except that the index k now refers to positions within the particular column under analysis. The results of this decomposition of process 84 are inherently downsampled by a factor of two (e.g., by keeping only even-numbered values), as discussed above.

Similarly, process 86 performs low-pass and high-pass decomposition of the results $\{d^{j-1}\}$ on a column-by-column basis. The decomposition of process 86 is a convolution in similar fashion as discussed hereinabove, except that the index k now refers to position within the column under analysis. The results of column-wise high-pass decomposition process 85 are also inherently downsampled by a factor of two, as noted above.

The results of processes 84, 86 then correspond to four components of the input dual sequence $\{c^j\}$, and are stored back into image memory 67 in process 88. These components are referred to as the LL, LH, HL, HH components. By way of explanation, the LL component consists of those sequence entries resulting from column-wise low-pass process 84 upon the row-wise low-pass results $\{c^{j-1}\}$ of process 78; the HL component consists of the results of column-wise low-pass process 84 upon the row-wise high-pass results $\{d^{j-1}\}$ of process 80. Similarly, the LH component consists of the results of column-wise high-pass process 86 upon the row-wise low-pass results $\{c^{j-1}\}$ of process 78, while the HH component

consists of the results of column-wise high-pass process 86 upon the row-wise high pass results $\{d^{j-1}\}$ of process 80. Figure 9 illustrates, with array 94, the arrangement of these four components in image memory 67, with A=LL, B=HL, C=LH, and D=HH.

Upon storing of the decomposed components back into image memory 67 in process 88, the input document image is ready for quantizing in process 50, as will be described hereinbelow. It should be apparent to those of ordinary skill in the art, particularly with reference to equations (32), (33), that this embodiment of the invention provides the significant advantage that spline-wavelet decomposition may be performed using spline and wavelet functions of finite support, and which involve only integer calculations. As such, the decomposition process of this embodiment of the invention may be done at very high speed, similar to a digital FIR filter, or alternatively with relatively primitive processing units, while maintaining high quality of the compressed document.

In addition, it will become apparent, from the description hereinbelow regarding decompression of the document, that reconstruction of the document may be made using the same spline and wavelet sequences $\{a_k\}$, $\{b_k\}$ as used in decomposition of the document. As such, reconstruction may also be performed solely with integer operations, and thus at high speed or with inexpensive processors.

(ii) Interpolatory wavelet decomposition

Referring back to Figure 2, according to an alternative embodiment of the invention, decomposition process 48 may be performed by way of interpolatory wavelets (shown as process 48b). As noted above, it is contemplated that interpolatory wavelet decomposition will be most useful where the compression and decompression speed is of the highest importance, at a cost of somewhat reduced reproductive quality.

According to this embodiment of the invention, the document image is decomposed by way of the centralized B-spline scaling function, together with its corresponding wavelet. Figure 10a illustrates the centralized linear B-spline function $\phi_L(x)$, which is defined as follows:

$$\phi_L(x) = \begin{cases} 1-|x| & -1 \le x \le 1 \\ 0 & otherwise \end{cases} \qquad (36)$$

This expression is a unit left shift of the cardinal B-spline of equation (1) above. The corresponding wavelet function $\psi_L(x)$ to centralized linear B-spline function $\phi_L(x)$ is as follows:

$$\psi_L(x) = \phi_L(2x-1) \qquad (37)$$

and is illustrated in Figure 10b. The wavelet function $\psi_L(x)$ has an interpolatory property, and as such is commonly referred to as an interpolatory, or interpolating, wavelet. Further information regarding

interpolatory wavelets may be found in Chui and Li, "Dyadic Affine Decompositions and Functional Wavelet Transforms", CAT Report 295 (Center for Approximation Theory, Texas A&M University, April 1993).

From the linear B-spline, scaling function $\phi_L(x)$ may be expressed as follows:

$$\phi_L(x) = \frac{1}{2}\phi_L(2x+1) + \phi_L(2x) + \frac{1}{2}\phi_L(2x-1) \tag{38}$$

then:

$$\phi_L(2x) = \phi_L(x) - \frac{1}{2}\psi_L(x+1) - \frac{1}{2}\psi_L(x) \tag{39}$$

Using the general wavelet decomposition formula of:

$$\phi(2x-k) = \sum_l a_{k-2l}\phi(x-l) + \sum_l b_{k-2l}\psi(x-l) \tag{40}$$

one may then readily show that, for the centralized linear B-spline scaling function $\phi_L(x)$ and its interpolatory wavelet $\psi_L(x)$:

$$\{a_n\} = \{. . ., 0, 1, 0, . . .\}$$

and $$\{b_n\} = \{. . ., 0, -\tfrac{1}{2}, 1, -\tfrac{1}{2}, 0, . . .\} \tag{41}$$

both over non-negative integers.

Process 48b of Figure 2 is thus performed identically as process 48a, described above relative to

Figure 8, except that the decomposition sequences $\{a_n\}$, $\{b_n\}$ of equations (41) are used in processes 78, 80, 84, 86. As such, the interpolatory wavelets of this embodiment of the invention provide the same advantages, in decomposition, as described hereinabove relative to the dual base wavelet technique of process 48a, in that integer operations may be utilized for such decomposition.

However, differently from the dual base technique described hereinabove, the use of interpolatory wavelets require the reconstruction sequences $\{p_k\}$, $\{q_k\}$ to be different from decomposition sequences $\{a_n\}$, $\{b_n\}$. From equation (39) above:

$$\phi_L(x) = \frac{1}{2}\phi_L(2x+1) + \phi_L(2x) + \frac{1}{2}\phi_L(2x-1) \tag{42}$$

As such, the reconstruction sequence $\{p_k\}$ for the interpolatory wavelets (the use of which will be described in detail hereinbelow) will have $p_{-1} = ½$, $p_0 = 1$, and $p_1 = -½$, with all other values of p equal to zero. Given the simple expression of equation (37) for $\psi_L(x)$, the reconstruction sequence $\{q_k\}$ will have only a single non-zero element, namely $q_1 = 1$, with all other values of q equal to zero.

<u>(b) Full-tree spline wavelet decomposition using spline-wavelet packets</u>

According to another alternative embodiment of the invention, specifically the third approach noted above, decomposition process 48 may be performed according to

another technique which will be referred to herein as "wavelet packets" or "spline wavelet packets". As will be apparent from the description hereinbelow, this decomposition (or compression) approach utilizes a full-tree decomposition, rather than the half-tree approach discussed above. Full-tree decomposition refers to decomposition in which both the low-pass and the high-pass (or band-pass) components are decomposed when going to the next level of decomposition.

The use of orthogonal wavelet packets in wavelet analysis is known, as indicated by Coifman, Meyer, et al., "Signal Processing and Compression with Wave Packets", (1992). However, these wavelet packets are not spline-based, and are not symmetric, and as such their use in fields such as image compression and decompression would introduce undesirable phase distortion.

According to this alternative embodiment of the invention, process 48c (Figure 2) performs decomposition of the document image by way of non-orthogonal, symmetric, spline-based wavelet packets. The mathematical theory behind this class of wavelets is presented in Chui and Li, "Nonorthogonal Wavelet Packets", SIAM J. Math. Anal., Vol. 24, No. 3, (SIAM, May 1993), pp. 712-738. According to the present invention, these spline-based wavelet packets are used to further decompose the results of data, corresponding to a document, that are decomposed to a first level by application of a scaling function and the corresponding wavelet function. This further decomposition provides higher compression ratios and higher quality picture reproduction, when compared against the dual-base technique (process 48a of Figure 2) described hereinabove.

In contrast to the decomposition approach illustrated in Figures 5a, 5b, where only a single branch (the $\{c_n^j\}$ branch of Figure 5a) is applied to subsequent decomposition stages, both of the low-pass and high-pass components are further decomposed according to this wavelet packet technique.

Figure 11a illustrates the decomposition "tree" according to this alternative embodiment of the invention. As is evident therefrom, two-dimensional nomenclature is required to indicate the decomposition component of interest, since components of the n+1 level can come from either low-pass or high-pass results from the prior level; the nomenclature used in the tree of Figure 11a identifies each component sequence $\{c^{m,n}\}$ as the m^{th} lowest frequency component above the spline component in the n^{th} level of decomposition. For example, sequence $\{c^{2,3}\}$ is the sequence of coefficients representative of the second lowest frequency component above spline component $\{c^{0,3}\}$ in the third level of decomposition.

Figure 11a illustrates that each component sequence $\{c^{m,n}\}$ is decomposed, in the next level of decomposition, into a low-pass and a high-pass component, each of which is itself decomposed in the following level of decomposition. In the tree of Figure 11a, the input coefficient sequence $\{c^{0,0}\}$ (corresponding to the input signal function f(x)) is convolved with scaling function coefficient sequence $\{a_n\}$ to produce a low-pass component coefficient sequence $\{c^{0,1}\}$, and is also convolved with wavelet function coefficient sequence $\{b_n\}$ to produce a high-pass component coefficient sequence $\{c^{1,1}\}$. For the next level of decomposition, low-pass component coefficient sequence $\{c^{0,1}\}$ is convolved with scaling function coefficient sequence $\{a_n\}$ to produce a second

level low-pass component coefficient sequence $\{c^{0,2}\}$, and is also convolved with wavelet function coefficient sequence $\{b_n\}$ to produce a high-pass component coefficient sequence $\{c^{1,2}\}$. To this point, the decomposition is similar to that described hereinabove relative to the dual-base decomposition and interpolatory wavelet decomposition.

In the next level, however, according to the wavelet packet technique according to this embodiment of the invention, the first level high-pass component coefficient sequence $\{c^{1,1}\}$, which is basically a wavelet component, is also convolved with wavelet function coefficient sequence $\{b_n\}$ to produce a low-pass component coefficient sequence $\{c^{2,2}\}$, and is also convolved with scaling function coefficient sequence $\{a_n\}$ to produce a high-pass component coefficient sequence $\{c^{3,2}\}$. One should note that the coefficient sequences applied to a previous high-pass component are reversed from that of the tree of Figure 4a, such that the high-pass component of a previous high-pass component is obtained by convolution with scaling function coefficient sequence $\{a_n\}$, and so that the low-pass component of a previous high-pass component is obtained by convolution with wavelet function coefficient sequence $\{b_n\}$. Additional levels of decomposition can then continue, for each of the component coefficient sequences $\{c^{m,n}\}$ so generated, until the desired compression ratio is obtained.

The basic spline scaling function and corresponding wavelet function utilized according to this alternative method of the invention is governed by a family of basis functions. This family includes a series of functions μ_i, where μ_0 is a spline scaling function and where each function μ_i (for $i > 0$) is a wavelet function. In the general case, where $\mu_0(x)$ is equal to the scaling function

$\phi(x)$, the first wavelet function μ_1 will equal the corresponding wavelet function $\psi(x)$, and the progression of wavelet functions will be as follows:

$$\mu_{2l}(x) = \sum_k p_k \mu_l (2x-k) \tag{43}$$

and

$$\mu_{2l+1}(x) = \sum_k q_k \mu_l (2x-k) \tag{44}$$

As a result, the wavelet packet basis functions μ can be generated iteratively, in the general sense, by using the scaling function $\phi(x)$ and the corresponding wavelet function $\psi(x)$. Accordingly, a component coefficient sequence $\{c_{m,n}\}$ at any level of decomposition, including the original input coefficient sequence $\{c^{0,0}\}$ can be obtained by the linear combination of wavelet packets.

By way of explanation, decomposition according to this alternative technique will now be described in the case of a one-dimensional function $f(x)$. First, as in the prior cases, the input function $f(x)$ is projected into the scaling function, or spline, space as follows:

$$f(x) = \sum_j c_j^{0,0} \mu_0 (2x-j) = \sum_j c_j^{0,0} \phi(2x-j) \tag{45}$$

The wavelet packet decomposition of $\{c^{0,0}\}$ is then governed by the following formulae:

$$c^{k,l} = \sum_k (z^{\lambda_k}_{2l-k})\, c^{\lfloor \frac{k}{2} \rfloor, l-1} \tag{46}$$

where:

$$\lambda_k = \begin{cases} 0 & \text{for even } k \\ 1 & \text{for odd } k \end{cases} \tag{47}$$

and where:

$$\{z^{\lambda_k}_n\} = \begin{cases} \{a_n\} & \text{for } \lambda_k = 0 \\ \{b_n\} & \text{for } \lambda_k = 1 \end{cases} \tag{48}$$

The coefficient sequences $\{a_n\}$, $\{b_n\}$ are, again, the spline and wavelet coefficient sequences determined by selection of the scaling function and wavelet function selected for the decomposition. According to this embodiment of the invention, the linear and cubic coefficient sequences $\{a_n\}$, $\{b_n\}$ preferred for use in the wavelet packet decomposition of digital documents are listed in Appendix A hereto. Decomposition is then performed according to the tree diagram of Figure 11a, to the extent desired.

As will be described in further detail hereinbelow, reconstruction of a decomposed representation is performed by convolution of high-pass components with a reconstruction coefficient sequence $\{q_k\}$ and low-pass components with a reconstruction coefficient sequence $\{p_k\}$, in full-tree form, as shown in Figure 11b. The reconstruction formula utilized in this operation is as follows:

$$c^{k,l-1} = \sum_{k} \{p_{l-2k} c^{2k,l} + q_{l-2k} c^{2k+1,l}\}$$

(49)

Referring now to Figures 12a, 12b, 13, and 14, process 48c in which a two-dimensional function f(x,y) representative of a document, expressed as an array of sample values in byte-packed PGM format, is decomposed according to the wavelet packet technique, will now be described in detail. As shown in Figure 12a, in process 96, DSP 60 projects the input data into spline space (as a coefficient sequence $\{c^{0,0}\}$). As described hereinabove, for a digital array of data as provided by format converter 12, the coefficient sequence corresponds to the values in the digital array f(x,y). The arrangement of input coefficient sequence $\{c_{0,0}\}$ in an array of coefficients {X} is shown in Figure 13a, which is at the beginning of the decomposition tree of Figure 14.

Decomposition of the input coefficient sequence $\{c_{0,0}\}$ (or array {X}) then begins with processes 98, 100, in which the sequence $\{c_{0,0}\}$ is convolved, in row-wise fashion, with decomposition coefficient sequences $\{a_n\}$, $\{b_n\}$ to produce the low-pass and high-pass component coefficient arrays {X}, {A}, respectively. Decomposition coefficient sequences $\{a_n\}$, $\{b_n\}$ are based upon scaling functions and wavelet functions suitable for use in this wavelet packet approach, such as those shown in Appendix A. Each convolution in the decomposition sequence contains downsampling by a factor of two, as in the cases discussed hereinabove. As such, as shown in Figure 13b, each row of the image data contains a row-wise low-pass component coefficient array {X} and a row-wise high-pass component coefficient array {A}. For purposes of description of Figures 13 and 14, the low-pass component

coefficient sequence will retain the designation of the sequence from which it was generated (hence array {X} in Figure 13b is the low-pass component of array {X} in preceding Figure 13a).

The result of process 98, namely the low-pass component of the input image data array is then again decomposed, in row-wise fashion, in processes 102, 103. Process 102 convolves coefficient array {X} of Figure 13b with spline decomposition coefficient sequence $\{a_n\}$, including downsampling, to generate another row-wise low-pass component, shown as coefficient array {X} in Figure 13c. Process 103 convolves coefficient array {X} of Figure 13b with wavelet decomposition coefficient sequence $\{b_n\}$, including downsampling, to generate a row-wise high-pass component, shown as coefficient array {B} in Figure 13c.

The high-pass results from process 100 are similarly decomposed again, in a row-wise direction, in processes 104, 105. Process 104 convolves the high-pass coefficient array {A} of Figure 13b with wavelet decomposition coefficient sequence $\{b_n\}$, including downsampling, to produce a low-pass coefficient array {A} of Figure 13c. Similarly, process 105 convolves high-pass coefficient array {A} of Figure 13b with spline decomposition coefficient sequence $\{a_n\}$, including downsampling, to produce a high-pass coefficient array {C} of Figure 13c. As noted hereinabove, according to this wavelet packet technique, the spline decomposition coefficient sequence $\{a_n\}$ and wavelet decomposition coefficient sequence $\{b_n\}$ are reversed, in this second decomposition pass when applied to the high-pass results of the first decomposition pass.

The results of processes 102, 103, 104, 105 (together corresponding to the arrangement illustrated in Figure 13c) are then each applied to column-wise decomposition processes 106LL, 106LH, 106HL, 106HH, respectively. Figure 12b generically illustrates the subprocess steps performed in each of column-wise decomposition processes 106LL, 106LH, 106HL, 106HH.

Referring to Figure 12b, subprocess 108 performs the low-pass decomposition of its corresponding components (see Figure 13c) in a column-by-column manner, by convolving the coefficient arrays with spline decomposition coefficient sequence $\{a_n\}$ in the manner discussed above, which includes downsampling by a factor of two. As a result, each of the coefficient arrays {X}, {B}, {A}, {C} of Figure 13c provide low-pass coefficient arrays {X}, {B}, {A}, {C}, respectively, as shown in Figure 13d. Subprocess 110 similarly convolves its corresponding coefficient array of Figure 13c with wavelet decomposition coefficient sequence $\{b_n\}$ and downsamples the result by a factor of two. As a result, each of the coefficient arrays {X}, {B}, {A}, {C} of Figure 13c thus also provide high-pass coefficient arrays {D}, {E}, {F}, {G}, respectively, as shown in Figure 13d.

The results of this first column-by-column decomposition step are then repeated for a second pass, as before. The low-pass results of subprocess 108 are convolved in a corresponding subprocess 111, in column-wise fashion, with spline coefficient sequence $\{a_n\}$, including downsampling; the cumulative effect of subprocess 111 for all of the results of processes 102, 103, 104, 105 is to produce low-pass coefficient arrays {X}, {B}, {A}, {C} of Figure 13d. Subprocess 112 convolves its corresponding high-pass coefficient array {X}, {B}, {A}, {C} of Figure 13d, in column-wise fashion,

with wavelet coefficient sequence $\{b_n\}$, with the results downsampled, to cumulatively produce high-pass coefficient arrays $\{H\}$, $\{J\}$, $\{L\}$, $\{N\}$, respectively. Similarly, each of the high-pass coefficient arrays $\{D\}$, $\{E\}$, $\{F\}$, $\{G\}$ of Figure 13d from subprocess 110 are convolved, in a corresponding subprocess 113, in column-wise fashion, with wavelet decomposition coefficient sequence $\{b_n\}$ (downsampled) to cumulatively produce low-pass component coefficient arrays $\{D\}$, $\{E\}$, $\{F\}$, $\{G\}$ shown in Figure 13e. Each of high-pass coefficient arrays $\{D\}$, $\{E\}$, $\{F\}$, $\{G\}$ of Figure 13d are also convolved in a corresponding subprocess 114, in column-wise fashion, with spline decomposition coefficient sequence $\{a_n\}$, including downsampling, to cumulatively produce high-pass component coefficient arrays $\{I\}$, $\{k\}$, $\{M\}$, $\{O\}$ of Figure 13e.

The results of subprocesses 111, 112, 113, 114 in each of processes 106LL, 106LH, 106HL, 106HH are then stored by process 116 in image memory 67, in the form illustrated in Figure 13d. Accordingly, the full-tree decomposition of Figure 14 is complete at this time, with the "blur" image of the input image sequence represented by the lowest-pass component of the row-wise and column-wise decomposition, represented by coefficient array $\{X\}$ of Figures 13e and 14. All of the other coefficient arrays $\{A\}$ through $\{O\}$ of Figure 13e represent wavelet packet components from this decomposition.

According to this alternative embodiment of the invention, the decomposition tree of Figure 13e, and in general the decomposition tree of Figure 11a, may be "pruned", or stopped, at any particular component. Referring to Figure 11a, it may be possible to stop the decomposition process for an intermediate decomposition component, for example at sequence $\{c^{2,2}\}$; in this

example, the remainder of the coefficient sequences in the decomposition would be obtained by convolution in the third level, except that neither of sequences $\{c^{5,3}\}$ and $\{c^{4,3}\}$ would be generated. This allows for a reduction in the amount of data to be stored, and processed in subsequent levels, particularly if no additional compression thereof is available. For example, a particular sequence result of a decomposition may have all values equal to one another; further decomposition would not provide any additional compression, particularly over that which a lossless compression would provide. Pruning of the decomposition tree at such a result would thus be advisable.

When pruning of the full decomposition tree according to the wavelet packet technique is used, however, the arrangement of data in the resulting array must also be indicated within the data stream sent to downstream processing units (i.e., decompression system 30), in order to communicate which components of the full tree decomposition have been retained. Referring now to Figure 15, an example of the data frame useful in communicating the results of wavelet packet decomposition will now be described. The data frame shown in Figure 15 corresponds to the format as stored in image memory 67 upon completion of the decomposition process (prior to the quantization, lossless compression, and other processes described hereinbelow), and may also correspond to the compressed image data as communicated over network 26 or as stored on disk 22, 24 (after quantization, lossless compression, and the other processes described hereinbelow).

As shown in Figure 15, data frame 118 consists primarily of a data portion 120, which contains the coefficient sequence results of the decomposition process

described above. Header 119 indicates the beginning of data frame 118, and contains conventional information regarding the length and identity of data frame 118, as does header portion 123; trailer 121 follows the data portion 120 to indicate the end of data frame 118, and may also contain identity information. Part of the overall header of data frame 118 (i.e., the information preceding the actual data portion 120) according to this embodiment of the invention is tree map 122, which contains information concerning the components contained within data portion 120, especially considering the availability of "pruning" of the decomposition tree allowed according to the full-tree wavelet packet decomposition method described hereinabove.

Tree map 122 includes a first word 124 that indicates the number of decomposition levels performed. In this embodiment of the invention, map words 126 are provided for each decomposition level to indicate which components are present at each level of decomposition; as such, word 124 is necessary to indicate the number of map words 126 to follow. Each map word 126 contains a number of characters (which may be single or plural bits) equal to the number of possible components for that level; in other words, the m^{th} level of decomposition has 2^m possible decomposition components. As such, map word 126_0 has one character as it corresponds to the zeroeth level of decomposition; it may therefore be possible, given the construction of header 119, 123, to eliminate map word 126_0 since it merely indicates the presence or absence of any data in data portion 120. First level map word 126_1 contains two characters, considering that the first level decomposition results in one low-pass component and one high-pass component. Second level map word 126_2 contains four (2^2) characters, third level map word 126_3 contains

eight (2^3) characters, and so on, for the number of decomposition levels performed.

Figure 16 illustrates the decomposition tree of Figure 14, pruned so that the sequence {A} of Figure 13c was not further decomposed, and that sequences {B} and {C} of Figure 13d were not further decomposed. If, for example, each character in each of map words 126 contains a "1" to indicate the presence of a component and contains a "0" to indicate the absence of a component because of pruning, the following table indicates the contents of map words 126 (left-most bit containing the character for the lowest frequency component):

zeroeth level :	126_0 =	1
first level :	126_1 =	11
second level :	126_2 =	1111
third level :	126_3 =	1111 0011
fourth level :	126_4 =	1111 0011 0000 0011

As a result of this construction of data frame 118, a map of the component sequences obtained by the full-tree decomposition 48c may be communicated along with the sequences. Of course, other techniques for communicating this information may alternatively be used.

As in the case of the previous examples of this embodiment of the invention, the operations required of DSP 60 in performing the wavelet packet decomposition of process 48c are integer operations, given the integer nature of the decomposition coefficient sequences $\{a_n\}$, $\{b_n\}$ of Appendix A hereto. As such, the complex operations of process 48c may still be performed relatively quickly, and by processors of moderate capability; of course, the wavelet packet decomposition (and reconstruction, as will be described hereinbelow)

requires additional computing resources compared to the dual base technique and the interpolatory wavelet techniques described hereinabove.

The wavelet packet technique provides a high degree of compression (by a factor of 16, for example) in each pass, as compared to the dual base technique and the interpolatory wavelet techniques. In addition, because of the full tree decomposition, the quality of the reproduction from the compressed data will be somewhat higher than the other techniques. As such, the wavelet packet technique described herein will be attractive for certain high quality applications, and as such is a useful option within compressor system 20 according to this embodiment of the invention.

iii. Completion of the compression operation

Regardless of the particular decomposition option selected as process 48 of Figure 2, the results of the decomposition operation, as stored in image memory 67, are then ready for the remainder of the compression process, as will now be described relative to Figure 2.

Referring back to Figure 2, process 50 is next performed by quantization processor 62 upon the results of the decomposition stored in image memory 67. Specifically, the wavelet, or high-frequency, components of the spline-wavelet decomposition process 48 (i.e., the LH, HL and HH components of the dual-base wavelet decomposition and the interpolatory wavelet decomposition, and the wavelet packet components of the wavelet packet decomposition) are subjected to thresholding and quantization in process 50. It is preferred that quantization is not performed upon the

"blur" components (i.e., LL, or {X}), as this component contains most of the significant information of the message, and as such any quantization and thresholding would cause loss of quality without a significant increase in the compression ratio.

However, quantization of the high-frequency wavelet components of the decomposition may readily be performed, since most real-world images in PGM format will consist primarily of low-frequency intensities, such that the higher-frequency wavelet components will tend to have a large number of small, or zero, coefficient values. According to the preferred embodiment of the invention, therefore, this large population of small values in the higher-frequency components after decomposition may are discarded as a result of the quantization of process 50. The memory requirements for storage of the coefficients that undergo the thresholding and quantization are thus much reduced, even before the application of lossless compression techniques as will be noted below.

As noted above, quantization processor 62 is a logic circuit for filtering the data corresponding to decomposed images in order to achieve the desired compression ratio, based upon one of a number of available quantization modes. The simplest technique for performing quantization is referred to as "thresholding", where any coefficient having an absolute value less than a certain threshold value is set to zero, and where all remaining coefficient values are rounded to the nearest integer value. The other quantization modes indicated above may instead be performed in process 50 by quantization processor 62, according to the known conventional techniques noted above.

Upon completion of the quantization of process 50, decision 51 is performed to determine if the desired compression ratio has yet been achieved. Upon the completion of a single decomposition process 48, the blur component of the image may be adequate to accurately convey the input image. This single decomposition of the image will provide up to a 4:1 compression ratio (for the dual-base and interpolatory wavelet case) or 16:1 ratio (for the wavelet packet case), depending upon the memory requirements for the quantized high frequency results. This maximum ratio is obtained if all higher frequency components are discarded in the quantization process 50, leaving only the blur component. Decision 51 thus determines if the compression ratio achieved so far is adequate for the desired transmission or storage and, if not, passes control back to the decomposition process 48 so that the blur component from the prior decomposition may again be decomposed according to the selected optional technique.

According to this embodiment of the invention, the determination of whether the desired compression ratio has been obtained may be done relative to a predetermined compression ratio. In such a case, data controller 56 will maintain a count of the number of passes through decomposition process 48, and will perform decision 51 by comparing the resulting compression ratio against a previously stored value.

Alternatively, the compression ratio decision 51 may be determined in a dynamic manner according to the accuracy with which the blur component is representative of the input frame image. Conceptually, such a determination will be a measure of the difference between the blur component and the input image block relative to a predetermined accuracy limit, such that if the blur

decomposition is within a preselected ϵ limit, an additional pass through the decomposition process 48 may be performed. It is contemplated that this determination can be made automatically by data controller 56 in decompressor 20 so that the process performed in each of channel compression subsystems may remain consistent, for example by calculating a numerical value based upon the sum of the coefficients in the high-frequency components of the decomposed image, which will indicate the difference between the input image to the blur image result of the decomposition process 48.

Whether statically or dynamically determined, upon decision 51 returning the result that the desired compression ratio has been obtained, according to the preferred embodiment of the invention, lossless compression is then performed upon the results of the decomposed and quantized images for the frame, in process 52 of Figure 2. Referring back to Figure 4, the lossless compression of process 52 is preferably performed by lossless compressor 64 according to a conventional lossless technique such as Huffman encoding. The lossless compression of process 52 is especially beneficial in compressing the quantized higher frequency components from the decomposition, considering that non-zero or varying values in these components will be quite sparse for most video image frames.

After the lossless compression of process 52, the compressed image data is then formatted for transmission or storage, as the case may be. It is preferred that the coding of the compressed image data be performed within compressor 20, preferably by data flow interface 66 shown in Figure 4, prior to its application to digital communications network 26 or bus 25. Besides the actual decomposition sequences, as compressed, the transmitted

compressed document image data must include information regarding the type of compression performed; such information must identify the type of lossless compression (if any), the spline-wavelet decomposition process, and the type of byte-packing and number of color bits, used in the compression process. It is contemplated that such coding, including the insertion of this necessary data control information and document identity information as a header or trailer. Examples of such information include x-dimension image size, y-dimension image size, the number of bits used in byte packing, the wavelet decomposition method used (i.e., dual base wavelet, interpolatory wavelet, wavelet packet), the number of levels of decomposition, the quantization method used, and the lossless compression method used. This and equivalent information and arrangements will be within the knowledge of one of ordinary skill in the art having reference to this specification.

3. The decompression system

The construction and operation of decompression system 30 according to the preferred embodiment of the invention will now be generally described. As in the case of compression system 10, decompression system 30 may be a stand-alone system, but is preferably arranged so as to be an add-on card for a conventional high performance personal computer or workstation or as a function of a larger computer, such as a mainframe computer or supercomputer. It is also contemplated, as will be apparent from the following description, that it will be possible to implement decompression system 30 with the same hardware as compression system 10, considering that decompression system 30 is performing

substantially the inverse operations of those described hereinabove, and as such uses quite similar circuitry to accomplish the same.

However, it is also contemplated that many applications will utilize a central compression system 10, with many remote decompression systems 30 able to receive or retrieve compressed image data therefrom. In such an arrangement, there is no need for the remote stations to have compression capability, as they will be "read-only" systems. Accordingly, the construction and operation of decompression system 10, and specifically spline-wavelet decompressor 40, as a decompress-only subsystem will now be provided.

As discussed above and as shown in Figure 1, decompression system 30 includes spline-wavelet decompressor 40 and format converter 32. Spline-wavelet decompressor 40 receives the compressed image data from DCN 26, or from disk storage 22, 24 via bus 27, and reconstructs the document image from the decomposition sequences in the manner to be described hereinbelow. Format converter 32 operates to reformat the decompressed document into the suitable form for display on video display 34d, printing by printer 34p, or output by way of another conventional output device.

Referring now to Figures 17 and 18, the construction and operation of spline-wavelet decompressor 40 to reconstruct, or decompress, the received or retrieved document image data according to the preferred embodiment of the invention will now be described in detail. Spline-wavelet decompressor 40 includes data controller 156 which controls the operation of the other components within spline-wavelet decompressor 40. In this example, data controller 156 presents and receives signals on

control bus 163 to control the timing, feedback and transmission of information through and from spline-wavelet decompressor 40. It is therefore contemplated that data controller 156 may be implemented as a relatively simple logic circuit, for example as implemented into a gate-array or other semi-custom logic circuit, for performing these functions.

Spline-wavelet compressor 40 also includes data flow interface 166, which receives a compressed document image data from DCN 26 or bus 27 in process 128 (Figure 17). Interface 166 provides an interface between spline-wavelet decompressor 40 and network 26 or bus 27, and as such is able to strip header and trailer information from the received data stream which, as noted above, may include information necessary for decompressor 40 to comprehend the compression techniques used in the compression of the received data. Interface 166 communicates this identity and control information contained therein to other components in decompressor 40, by way of various control lines, some of which are shown in Figure 18.

After proper receipt and reformatting of the received document data by interface 166, lossless decompression process 130 is next performed by lossless decompressor 164. Lossless decompressor 164 may be implemented by way of a conventional digital signal processor such as the TMS320C40 or TMS320C30 available from Texas Instruments Incorporated, the i860 processor available from Intel Corporation, or general purpose microprocessors such as the 80386 and 80486 available from Intel Corporation or the 68030 and 68040 available from Motorola, programmed in such a manner as to perform lossless decompression process 130 upon the data received from interface 166; alternatively, lossless compressor 64

may be implemented as a custom logic circuit for providing this function. Lossless decompression process 130 will be performed by lossless decompressor 164 according to the technique used in the compression of the data (as communicated thereto by interface 166 from the header data). As noted above, these lossless decompression techniques include Huffman encoding, adaptive Huffman encoding, arithmetic encoding, LSQ coding, and the like. The output of lossless compressor 64 is forwarded to dequantization processor 162, for dequantization in process 132.

According to this embodiment of the invention, dequantization processor 62 is preferably implemented as a programmable microprocessor or custom logic circuit for performing the functions described hereinbelow; such implementation is believed to be readily apparent to one of ordinary skill in the art having reference to this description. As shown in Figure 18, dequantization processor 162 is connected to image memory banks 167 by way of data bus 161d and address bus 161a, so that dequantization processor 162 can store the results of dequantization process 132 therein, and retrieve data therefrom in the event that additional reconstruction is performed. In this example, as in the case of compressor 20, image memory 167 is arranged in four banks, totaling two megabytes in capacity. In process 132, dequantization processor 162 dequantizes the information in process 132 and stores the results in image memory 167, awaiting spline-wavelet reconstruction.

Dequantization process 132 is performed by dequantization processor 62 according to the quantization mode selected in the compression process for the particular document, as noted above. The particular quantization mode used in the compression is communicated

to dequantization processor 162 by interface 166, based upon information communicated in the document data stream. As noted above, conventional quantization methods that may be used in the compression process (the inverse of which is performed in process 132 of Figure 17) include no quantization, simple thresholding, scalar quantization, JPEG quantization using tables, the Federal Bureau of Investigation quantization standard for fingerprint compression, and others.

As noted above, upon completion of dequantization process 132 by dequantization processor 162, the results of the dequantization are stored in image memory 167 to await reconstruction by digital signal processor (DSP) 160. DSP 160 is the main processing unit for performing spline-wavelet reconstruction process 148 of Figure 17. Examples of modern digital signal processors suitable for use as DSP 160 according to this embodiment of the invention are the TMS320C25 and TMS320C30 digital signal processors manufactured and sold by Texas Instruments Incorporated; of course, other digital signal processors and microprocessors may alternatively be used to perform the spline-wavelet decomposition operations described hereinbelow. DSP 160 is coupled to data bus 161d and address bus 161a (collectively referred to as memory bus 161) for communication with program memory 165 and image memory 167. According to this embodiment of the invention, and similarly as in the case of data compression described hereinabove, DSP 160 is programmed by way of code stored within DSP 160 or in program memory 165, to perform decomposition of the compressed document data in image memory 167 according to pre-calculated reconstruction coefficients stored in program memory 165. During and after reconstruction process 134, the document image data is stored in image memory 167. DSP 160 is also connected to control bus 163, so that it may be

controlled by data controller 156 and so that it can effect the necessary control of other components in spline-wavelet decompressor 40, including access of memory 165, 167, during the performance of decomposition process 134.

As noted above, according to this embodiment of the invention, spline-wavelet reconstruction process 134 will be performed by DSP 160 according to the decomposition or compression technique (i.e., dual base wavelet decomposition, interpolatory wavelet decomposition, or wavelet packet decomposition) that was used to compress the document image data received by decompressor 40; the decomposition mode so used will be, as noted above, communicated as part of the data stream and identified by data interface 166. The three reconstruction processes 134a, 134b, 134c will be described in detail hereinbelow.

a. Dual base wavelet reconstruction

Reconstruction process 134a reconstructs a document from coefficient sequences generated by dual base wavelet decomposition process 48a described hereinabove. According to this embodiment of the invention, therefore, process 134a consists of substantially the reverse process of process 48a described hereinabove, with high-pass and low-pass coefficient sequences convoluted with reconstruction sequences $\{p_k\}$, $\{q_k\}$, respectively. Because of the use of the dual scaling function $\phi_m(x)$ and dual wavelet function $\psi_m(x)$, the reconstruction sequences $\{p_k\}$, $\{q_k\}$ exactly equal the decomposition sequences $\{a_n\}$, $\{b_n\}$, respectively. As a result, the reconstruction process 134a may also be done purely with integer operations, thus facilitating the decompression process

by modern data processing hardware of reasonable performance levels.

As indicated from the description of the decomposition process hereinabove, the compressed document image data provided by dequantization process 132 contains coefficient sequences corresponding to the "LL", "LH", "HL" and "HH" decomposition components, from a particular decomposition level j. If, for example, the decomposition of process 48a for a particular document was done with a single pass (row convolution and column convolution), the coefficient sequences "LL", "LH", "HL" and "HH" stored in image memory 167 would correspond to array 94 of Figure 9 discussed hereinabove.

Referring now to Figure 19, reconstruction process 134a will now be described in detail. For the first pass through process 134a, these coefficient sequences will be in dual space, as noted above. Accordingly, process 134a begins with process 140 which, prior to spline-wavelet reconstruction, first transforms the coefficient sequences "LL", "LH", "HL" and "HH" from dual space into spline space. Process 140 is preferably performed by way of the well-known Cholesky computation method, given the duality principle as discussed above relative to equations (20) et seq. Subsequent passes through process 134a (i.e., when the result of decision 135, described hereinbelow, indicates that the document was not fully reconstructed in a prior pass through process 134a) will not perform the transformation of process 140, as the coefficient sequences will already have been transformed into spline space.

Upon completion of (or skipping) process 140, process 134a begins performing the inverse of decomposition process 48a described hereinabove. The

reconstruction of process 134a in effect operates according to the reconstruction tree of Figure 5b, in which the document component coefficient sequences are convolved with reconstruction coefficient sequences $\{p_k\}$, $\{q_k\}$. Specifically, the following reconstruction formula will apply:

$$c_k^{j+1} = \sum_l \left(p_{k-2l} c_l^j + q_{k-2l} d_l^j \right) \tag{50}$$

As noted above, the convolution processes in decomposition process 48a inherently downsample the results by operation of the summation limits; similarly, the convolution processes in reconstruction process 134a results in an upsampling of the results by operation of equation (50).

Since the last convolutions in process 48a were taken in the columnar direction for the two-dimensional sequences, the first convolution processes 142, 144 in process 134a are also taken in the column-wise direction. Process 142 convolves, in the column-wise direction, the "LL" and "LH" components with reconstruction coefficient sequence $\{p_k\}$, which equals decomposition sequence $\{a_k\}$ used in process 48a hereinabove. As noted in equation (32) for the linear case, this sequence $\{a_k\}$ is defined as:

$$\{a_0, a_1, a_2\} = \frac{1}{4}\{1, 2, 1\}, \qquad a_k = 0 \ for\ all\ other\ k \tag{32}$$

The 1/4 term may be applied to the resultant data later, allowing the convolution of process 142 to be performed solely with integer values. In addition, the coefficient sequences "LL" and "LH" are inherently upsampled by a factor of two in the convolution of process 142, as noted above. The results of the two convolution operations of process 142 are summed into coefficient sequence $\{c^{j+1}\}$.

Similarly, process 144 is performed by way of which the "HL", "HH" component coefficient sequences are convolved, in column-wise fashion, with the reconstruction coefficient sequence $\{q_k\}$; as noted above, reconstruction coefficient sequence $\{q_k\}$ exactly equals decomposition sequence $\{b_k\}$ used in the decomposition of the input document. For the linear case, the sequence $\{b_k\}$ is specified in equation (33) as follows:

$$\{b_0, b_1, b_2, b_3, b_4\} = \frac{1}{24}\{1, -6, 10, -6, 1\}, \quad b_k = 0 \textit{ for all other } k \tag{33}$$

Again, the 1/24 division may be performed at a later time, to allow the convolution of process 144 to be performed using only integer values. This convolution will also effectively upsample the coefficient sequences as noted above, and the sum of the results of the two convolution operations of process 144 may be considered as coefficient sequence $\{d^{j+1}\}$.

The coefficient sequences $\{c^{j+1}\}$, $\{d^{j+1}\}$ produced by the column-wise convolution of processes 142, 144 are then stored in image memory 167, in process 145. Referring back to Figure 9 for the example of a single pass decomposition and reconstruction, the coefficient

sequences $\{c^{j+1}\}$, $\{d^{j+1}\}$ at this stage correspond to array 92.

Processes 146, 148 are then performed, by way of which the column-wise convolution results $\{c^{j+1}\}$, $\{d^{j+1}\}$ are convolved in the row-wise direction. Process 146 convolves sequence $\{c^{j+1}\}$ in the row-wise direction with decomposition/reconstruction coefficient sequence $\{a_k\}$, while process 148 convolves sequence $\{d^{j+1}\}$ in the row-wise direction with decomposition/reconstruction coefficient sequence $\{b_k\}$. As in the prior convolution processes, an inherent upsampling occurs in these processes, due to the selected summation levels. The result of the convolution processes 146, 148, when summed, is the coefficient sequence $\{c^j\}$, which is then stored in image memory 167 in process 150. The resultant sequence $\{c^j\}$ at this stage of reconstruction, for the example of a single pass decomposition and reconstruction (i.e., j=0) corresponds to array 90 of Figure 9. Decision 135 (Figure 17) is then performed to determine if the value j at this point is equal to zero, in which case the document image has been fully reconstructed consistent with the information of the compression header described hereinabove, and processing will continue in the manner described hereinbelow.

b. Interpolatory wavelet reconstruction

As indicated hereinabove, reconstruction process 134b is used in the decompression of documents that were compressed according to the interpolatory wavelet decomposition process 48b described hereinabove. Also as described hereinabove, interpolatory wavelet decomposition process 48b is substantially identical to dual base wavelet decomposition process 48a, except that

the decomposition sequences $\{a_n\}$, $\{b_n\}$ of equations (41) are used, incorporating the interpolatory spline function and its associated wavelet.

As indicated hereinabove, and in contrast to the dual base wavelet reconstruction process 134b, the reconstruction coefficient sequences $\{p_k\}$, $\{q_k\}$ for the interpolatory approach are different from the decomposition sequences $\{a_k\}$, $\{b_k\}$. Based on equation (42) described hereinabove, the reconstruction sequence $\{p_k\}$ for the interpolatory wavelets will have $p_{-1} = ½$, $p_0 = 1$, and $p_1 = -½$, with all other values of p equal to zero; and based on equation (37), the reconstruction sequence $\{q_k\}$ will have only a single non-zero element, namely $q_1 = 1$, with all other values of q equal to zero.

Figure 20 illustrates the operation of reconstruction process 134b according to the interpolatory wavelet technique. Process 152 retrieves coefficient sequences "LL", "LH", "HL" and "HH" from image memory 167; unlike process 134a, however, no transformation from dual space into spline space is required in this case, as no transformation into dual space was performed in the decomposition.

The reconstruction of process 134b also operates according to the reconstruction tree of Figure 5b, in which the document component coefficient sequences are convolved with the reconstruction coefficient sequences $\{p_k\}$, $\{q_k\}$ defined hereinabove, according to the reconstruction formula of equation (50) hereinabove. In addition, just as the convolutions in decomposition process 48 inherently downsampled the results by operation of the summation limits; similarly, the convolution processes in reconstruction process 134b effectively upsample by operation of equation (50).

Similarly as in the case of process 134, process 134b first convolves the in the columnar direction for the two-dimensional sequences. Process 154 convolves, in the column-wise direction, the "LL" and "LH" components with reconstruction coefficient sequence $\{p_k\}$ which, in the linear case, equals $p_{-1} = ½$, $p_0 = 1$, and $p_1 = -½$, with all other values of p equal to zero. The 1/2 term may be applied to the resultant data later, allowing the convolution of process 154 to be performed solely with integer values. The (upsampled) results of the two convolution operations of process 154 are summed into coefficient sequence $\{c^{j+1}\}$. Similarly, process 155 is performed by way of which the "HL", "HH" component coefficient sequences are convolved, in column-wise fashion, with the reconstruction coefficient sequence $\{q_k\}$ which, in the linear case, equals $q_1 = 1$, with all other values of q equal to zero. The convolution of process 144 may thus be performed using only integer values. This convolution will also effectively upsample the coefficient sequences as noted above, and the sum of the results of the two convolution operations of process 155 may be considered as coefficient sequence $\{d^{j+1}\}$.

The coefficient sequences $\{c^{j+1}\}$, $\{d^{j+1}\}$ produced by the column-wise convolution of processes 154, 155 are then stored in image memory 167, in process 157. Processes 159, 161 are then performed, by way of which the column-wise convolution results $\{c^{j+1}\}$, $\{d^{j+1}\}$ are convolved in the row-wise direction. Process 159 convolves sequence $\{c^{j+1}\}$ in the row-wise direction with reconstruction coefficient sequence $\{p_k\}$, while process 161 convolves sequence $\{d^{j+1}\}$ in the row-wise direction with reconstruction coefficient sequence $\{q_k\}$. As in the prior convolution processes, an inherent upsampling occurs in these processes, due to the selected summation levels. The result of the convolution processes 159,

161 when summed, is the coefficient sequence $\{c_j\}$, which is then stored in image memory 167 in process 163. Decision 135 (Figure 17) is then performed to determine if the value j at this point is equal to zero, in which case the document image has been fully reconstructed and processing will continue in the manner described hereinbelow.

c. Wavelet packet reconstruction

For documents that were decomposed and compressed by way of the full-tree decomposition using wavelet packets of process 48c, reconstruction of the compressed document is performed by way of reconstruction process 134c which also operates according to wavelet packets. Wavelet packet reconstruction is governed by equation (49):

$$C^{k,l-1} = \sum_{k} \left(p_{l-2k} C^{2k,l} + q_{l-2k} C^{2k+1,l} \right) \tag{49}$$

The full tree reconstruction process, for two-dimensional data, thus will follow in the manner illustrated in Figure 21, which is the reverse of the full-tree decomposition of Figure 11a. As before, the reconstruction of components into a high-frequency component reverses the spline and wavelet sequences, so that, in reconstructing a high pass component in the next level, the high-pass component will be convolved with spline function coefficient reconstruction sequence $\{p_k\}$ and the low-pass component will be convolved with wavelet function coefficient reconstruction sequence $\{q_k\}$. Similarly, reconstruction process 134c is the reverse of decomposition process 134c described hereinabove.

According to this embodiment of the invention, and for the wavelet packet decomposition sequences listed in Appendix A hereto, spline function coefficient reconstruction sequence $\{p_k\}$ and wavelet function coefficient reconstruction sequence $\{q_k\}$ are listed in Appendix B hereto.

Referring now to Figures 22 and 23, reconstruction process 134c according to this embodiment of the invention will now be described in detail; further reference is directed to Figures 13a through 13e for the array arrangement of the various wavelet packet components of the document image. Reconstruction begins with process 170, in which the tree map (Figure 15) for the particular document is read and interpreted; in this way, decompressor 40 will operate only upon those components of the document image that are present, and can skip those components that are missing from the compressed document. Process 172 is then performed, in which DSP 160 retrieves component sequences from image memory 167 for convolution with reconstruction sequences $\{p_k\}$, $\{q_k\}$.

For purposes of explanation, reconstruction of the sequences of Figures 13a through 13e and 14 will now be described, relative to the flow diagram of Figure 22 and the reconstruction tree of Figure 23. This example will be described in the case where all components are present (i.e. no pruning of the decomposition tree was done); of course, if the tree map read in process 170 indicates that certain components were not produced in decomposition process 48c, those components will not be contemplated in the reconstruction.

Processes 171, 173, 175, 177 initiate the reconstruction by convolving the components in column-

wise fashion. Specifically, process 171 convolves components {X} (the blur image), {B}, {A}, and {C} with spline reconstruction coefficient sequence $\{p_k\}$ in column-wise fashion, and process 173 convolves components {H}, {J}, {L}, {N} with wavelet reconstruction coefficient sequence $\{q_k\}$ in column-wise fashion. Process 174 sums, in paired fashion, the results of processes 171, 173 to complete the upper-half (lower frequency) portion of the array in Figure 13d; convolved components {X} and {H} are summed into component {X} of Figure 13d, convolved components {B} and {J} are summed into component {B} of Figure 13d, convolved components {A} and {L} are summed into component {A} of Figure 13d, and convolved components {C} and {N} are summed into component {C} of Figure 13d.

As described above in decomposition process 48c, higher-frequency components were decomposed by convolution of the spline and wavelet sequences in reverse order. Reconstruction process 134c thus must reconstruct higher-frequency components by convolution of their respective low- and high-frequency components with the reconstruction sequences also in reverse order. Process 175 thus convolves components {D}, {E}, {F}, and {G} with wavelet reconstruction coefficient sequence $\{q_k\}$ in column-wise fashion, and process 177 convolves components {I}, {K}, {M}, {O} with spline reconstruction coefficient sequence $\{p_k\}$ in column-wise fashion. Process 174 sums, in paired fashion, the results of processes 175, 172 to complete the lower-half (higher frequency) portion of the array in Figure 13d. As such, convolved components {D} and {I} are summed into component {D} of Figure 13d, convolved components {E} and {K} are summed into component {E} of Figure 13d, convolved components {F} and {M} are summed into component {F} of Figure 13d, and convolved components {G} and {O} are summed into

component {G} of Figure 13d. Upon completion of process 174, the array of Figure 13d is reconstructed.

The next step of reconstruction is performed by a second set of column-wise convolutions in processes 176, 178. Process 176 convolves, column-wise, each of lower-frequency components {X} and {A} and higher-frequency components {E} and {G} with spline reconstruction coefficient sequence $\{p_k\}$, while process 178 column-wise convolves each of lower-frequency components {B} and {C} and higher-frequency components {D} and {F} with wavelet reconstruction coefficient sequence $\{q_k\}$. In process 180, the results of processes 176, 178 are summed in paired fashion, with convolved components {X} and {D} summed into component {X} of Figure 13c, convolved components {B} and {E} summed into component {B} of Figure 13c, convolved components {A} and {F} summed into component {A} of Figure 13c, and convolved components {C} and {G} summed into component {C} of Figure 13c.

Row-wise convolution operations are now to be performed to reconstruct the original image. Process 181 convolves component sequence {X} (the blur image) in row-wise fashion with spline reconstruction coefficient sequence $\{p_k\}$ and process 183 convolves component sequence {B} in row-wise fashion with wavelet reconstruction coefficient sequence $\{q_k\}$. Process 186 sums the results of processes 181, 183 into component sequence {X} of Figure 13b (i.e., the left-half-array). Similarly, process 185 convolves component sequence {A} in row-wise fashion with wavelet reconstruction coefficient sequence $\{q_k\}$ and process 187 convolves component sequence {c} in row-wise fashion with spline reconstruction coefficient sequence $\{p_k\}$. Process 184 sums the results of processes 185, 187 into component sequence {A} of Figure 13b (i.e., the right-half-array).

The reconstruction operation of process 134c is completed, for this level of reconstruction, by the row-wise convolution of component sequence {X} with spline reconstruction coefficient sequence $\{p_k\}$ in process 186, and by the row-wise convolution of component sequence {A} with wavelet reconstruction coefficient sequence $\{q_k\}$ in process 188. Process 190 sums the results of processes 186, 188 into the document image {X}, and stores these results in image memory 167.

d. Completion of reconstruction

Referring back to Figure 17, upon completion of the selected one of reconstruction processes 134a, 134b, 134c, decision 135 is performed under the direction of data controller 156 to determine if the document has been fully reconstructed according to the header compression information described hereinabove. For example, if the results of the completed process 134a, 134b, 134c returns a document that has a compression ratio, relative to the input document, of 1:1, spline-wavelet reconstruction process 134 is complete and decision 135 will pass control to process 136. In contrast, if in compression the decomposition process 48 was repeated to further compress the document, and thus if the compression ratio of the document image at the end of process 134 (e.g., at the first pass therethrough) relative to the input document is greater than 1:1, spline-wavelet reconstruction is not complete, and decision 135 will return control to dequantization process 132 for repetition of the appropriate spline-wavelet process 134.

Upon full reconstruction being reached (decision 135 returning a YES result), DSP 160 passes the results to

data controller 156 for presentation to format converter 32 of decompression system 30 (Figure 1). Format converter 32 is similarly constructed as format converter 12 discussed hereinabove, to perform the inverse operations of byte expansion and formatting for display. As discussed above, the document image data may have been byte-packed prior to spline-wavelet decompression, in which case byte-expansion process 136 will be necessary to recover the document. As such, format converter 32 will perform the necessary operations, according to information transmitted with the data that indicates the type of byte-packing performed and the numbers of bits per pixel of data that are to represent the decompressed document. Format converter 138 also performs format conversion process 138, in which the decompressed document is reformatted for display in the manner appropriate for the particular output medium (e.g., graphics display, printer output, disk storage in decompressed form, etc.). The reconstruction process is then complete, and the data may be presented in the desired manner.

4. Conclusion

The methods and systems for compressing and decompressing document image data described hereinabove relative to the present invention provide important advantages, as noted throughout the foregoing specification. These advantages include the fundamental benefits of wavelet analysis, where time-frequency localization of the input signal is implemented so that the time window narrows with increasing frequency and widens with decreasing frequency, thus providing highly accurate analysis for transient periods of the signal.

In addition, the document image compression and decompression techniques according to the preferred embodiments of the invention, in using spline-wavelet functions that have compact support, avoid boundary effect artifacts in the compressed documents. By avoiding boundary effects, the present invention greatly improves the accuracy of the reconstructed compressed document (relative to the input document), and also enables parallel processing (or computing) to be applied in the decomposition process by cutting the document into small pieces and, after reconstruction, piecing together the full image. Using the B-spline as the scaling function also provides display functions such as magnification and compression ("zoom-in" and "zoom-out") in the display of the reconstructed document. The usefulness of the reconstructed document, and thus the entire system, in allowing interactive display of the received or retrieved document is therefore much improved over conventional document compression and decompression systems.

In addition, it will have been appreciated that the present invention allows the compression and reconstruction operations to be done strictly with integer operations. As such, both compression and reconstruction operations according to the present invention may be done more efficiently, for a given compression ratio, than can conventional document compression and reconstruction processes. The ability to use integer operations also enables these processes to be performed with moderate performance data processing equipment, thus allowing wide-spread use of the present invention, as high-performance workstations are not necessarily required to practice the present invention.

Further, the present invention allows for several choices of compression and decompression routines, depending upon the desired tradeoff between processing complexity and document quality. It is therefore contemplated that the present invention will provide the industry with great flexibility in its implementation.

While the invention has been described herein relative to its preferred embodiments, it is of course contemplated that modifications of, and alternatives to, these embodiments, such modifications and alternatives obtaining the advantages and benefits of this invention, will be apparent to those of ordinary skill in the art having reference to this specification and its drawings. It is contemplated that such modifications and alternatives are within the scope of this invention as subsequently claimed herein.

APPENDIX A

	m = 2		m = 4	
k	a_k	b_{k+1}	a_{k+1}	$b_k + 1$
1	0.683012701892	0.866025403784	0.893162856314	-1.475394519892
2	0.316987298108	-0.316987298108	0.400680825467	0.468422596633
3	-0.116025403784	-0.232050807569	-0.282211870811	0.742097698477
4	-0.084936490539	0.084936490539	-0.232924626134	-0.345770890775
5	0.031088913246	0.062177826491	0.1290983571218	-0.389745580800
6	0.0222758664048	-0.022758664047	0.126457446356	0.196794277304
7	-0.008330249198	-0.016660498395	-0.066420837387	0.207690838380
8	-0.006098165652	0.006098165652	-0.067903608499	-0.106775803373
9	0.002232083545	0.004464167091	0.035226101674	-0.111058440711
10	0.001633998562	-0.001633998561	0.036373586989	0.057330952254
11	-0.000598084983	-0.001196169967	-0.018815686621	0.059433388390
12	-0.000437828595	0.000437828595	-0.019473269356	-0.030709700871
13	0.000160256388	0.000320512777	0.010066747520	-0.031811811318
14	0.000117315818	-0.000117315818	0.010424052187	0.016440944687
15	-0.000042940569	-0.000085881139	-0.005387929819	0.017028029466
16	-0.000031434679	0.000031434678	-0.005579839208	-0.008800839839
17	0.000011505891	0.000023011782	0.002883979478	-0.009114745138
18	0.000008422897	-0.000008422897	0.002986784625	0.004710957034
19	-0.000003082990	-0.000006165980	-0.001543728719	0.004878941541
20	-0.000002256905	0.0000022569054	-0.001598768083	-0.002521687975
21	0.000000826079	0.0000016521587	0.000826326663	-0.002611601542

APPENDIX B

	m = 2		m = 4	
k	p_k	q_{k+1}	p_{k+1}	q_{k+4}
1	$\frac{2}{2}$	$\frac{20}{4!}$	$\frac{6}{8}$	$-\frac{24264}{8!}$
2	$\frac{1}{2}$	$-\frac{12}{4!}$	$\frac{4}{8}$	$\frac{18482}{8!}$
3		$\frac{2}{4!}$	$\frac{1}{8}$	$-\frac{7904}{8!}$
4				$\frac{1677}{8!}$
5				$-\frac{124}{8!}$
6				$\frac{1}{8!}$

WHAT IS CLAIMED IS:

1. A method of compressing digital data representative of a document, comprising the steps of:

formatting digital data corresponding to an image of the document into a two-dimensional array of digital values, each value corresponding to the appearance of the document at a corresponding location of the document;

performing a first decomposing of the array of digital values, in a first direction, into low-frequency and high-frequency portions using a decomposition scaling function and a decomposition wavelet function, respectively, said decomposition scaling and wavelet functions not requiring support outside of the array;

performing a second decomposing of the results of the first decomposing step in a second direction, using the decomposition scaling and wavelet functions; and

communicating the results of the decomposing steps to a receiving unit.

2. The method of claim 1, wherein said communicating step comprises:

storing the results of the decomposing steps in a memory.

3. The method of claim 1, wherein said communicating step comprises:

transmitting the results of the decomposing steps to a decompression system.

4. The method of claim 3, further comprising:

after the transmitting step, operating the decompression system to perform a first reconstructing of the transmitted results in the second direction, using a reconstruction scaling function and a reconstruction wavelet function, said scaling and wavelet functions not requiring support outside of the array;

reconstructing the results of the first reconstructing step in the first direction, using the reconstruction scaling function and a reconstruction wavelet function; and

generating a visual output of the results of the reconstructing steps.

5. The method of claim 4, wherein the reconstruction scaling function is equal to the decomposition scaling function, and wherein the reconstruction wavelet function is equal to the decomposition wavelet function.

6. The method of claim 1, further comprising:

projecting the two-dimensional array of digital values into a dual space, the dual space having a dual decomposition scaling function corresponding to the decomposition scaling function, and having a dual decomposition wavelet function corresponding to the decomposition wavelet function;

and wherein said first and second decomposing steps are performed using the dual decomposition scaling function and the dual decomposition wavelet function.

7. The method of claim 1, wherein the decomposition scaling function is a centralized B-spline scaling function;

and wherein the decomposition wavelet function is the wavelet function corresponding to the centralized B-spline scaling function.

8. The method of claim 1, wherein the decomposition scaling function and decomposition wavelet functions are based upon non-orthogonal, symmetric, spline-based wavelet packets.

9. The method of claim 1, wherein the first decomposing step comprises:

(a) decomposing the array of digital values, in the first direction, into low-frequency and high-frequency portions using the decomposition scaling function and the decomposition wavelet function, respectively;

(b) decomposing the low-frequency portion from step (a), in the first direction, into low-frequency and high-frequency portions using the decomposition scaling function and the decomposition wavelet function, respectively; and

(c) decomposing the high-frequency portion from step (a), in the first direction, into low-frequency and high-frequency portions using the decomposition wavelet function and the decomposition scaling function, respectively.

10. The method of claim 9, wherein the second decomposing step comprises:

(d) decomposing the results of the first decomposing step, in the second direction, into low-frequency and high-frequency portions using the decomposition scaling function and the decomposition wavelet function, respectively;

(e) decomposing the low-frequency portion from step (d), in the second direction, into low-frequency and

high-frequency portions using the decomposition scaling function and the decomposition wavelet function, respectively; and

(f) decomposing the high-frequency portion from the step (e), in the second direction, into low-frequency and high-frequency portions using the decomposition wavelet function and the decomposition scaling function, respectively.

11. The method of claim 1, wherein the results of the second decomposing step comprise:

an LL component, corresponding to the low-frequency component, taken in the second direction, of the low-frequency decomposition in the first direction of the first decomposing step;

an LH component, corresponding to the high-frequency component, taken in the second direction, of the low-frequency decomposition in the first direction of the first decomposing step;

an HL component, corresponding to the low-frequency component, taken in the second direction, of the high-frequency decomposition in the first direction of the first decomposing step; and

an HH component, corresponding to the high-frequency component, taken in the second direction, of the high-frequency decomposition in the first direction of the first decomposing step;

and further comprising:

quantizing the LH, HL, and HH components prior to said communicating step.

12. The method of claim 5, further comprising:

after said quantizing step and prior to said communicating step, performing lossless compression on the LL, LH, HL, and HH components.

13. The method of claim 1, wherein said formatting step comprises:

substituting at least first and second digital values for display attributes corresponding to the image of the document, to generate a substituted array of the at least first and second digital values; and

after said substituting step, byte-packing the substituted array.

14. The method of claim 13, wherein said substituting step first, second and third digital values for display attributes corresponding to the image of the document.

15. A method of displaying compressed document image data, comprising:

receiving compressed image data corresponding to a document and storing the compressed image data in memory, said compressed image data arranged as blur and high-frequency coefficient sequences;

in a first image direction, applying a reconstruction scaling function and a reconstruction wavelet function to the compressed image data to reconstruct a low-frequency component and a high-frequency component;

in a second image direction, applying the reconstruction scaling function and the reconstruction wavelet function to the reconstructed low-frequency component and high-frequency component to reconstruct the document image;

displaying the reconstructed document image.

16. The method of claim 15, further comprising:

repeating said steps of applying the reconstruction scaling function and the reconstruction

wavelet function in the first and second image directions.

17. The method of claim 15, further comprising:

performing lossless decompression on the compressed document image data, prior to said applying steps.

18. The method of claim 15, wherein the applying steps are performed using integer operations.

19. The method of claim 15, further comprising:

transforming the compressed image data from dual space into spline space;

and wherein said first and second decomposing steps are performed using the dual decomposition scaling function and the dual decomposition wavelet function.

20. The method of claim 15, wherein the reconstruction scaling function is a centralized B-spline scaling function;

and wherein the reconstruction wavelet function is the wavelet function corresponding to the centralized B-spline scaling function.

21. The method of claim 15, wherein the compressed image data is contained within first and second pluralities of coefficient sequences;

wherein the step of applying the reconstruction scaling and wavelet functions in a first direction comprise:

convolving the reconstruction scaling function with the first plurality of coefficient sequences, in the first direction;

convolving the reconstruction wavelet function with the second plurality of coefficient sequences, in the first direction;

after the convolving steps, summing each of the first plurality of coefficient sequences to an associated one of the second plurality of coefficient sequences, to produce third and fourth pluralities of coefficient sequences;

after the summing step, convolving the reconstruction scaling function with the third plurality of coefficient sequences, in the first direction;

convolving the reconstruction wavelet function with the fourth plurality of coefficient sequences, in the first direction; and

then summing each of the third plurality of coefficient sequences to an associated one of the fourth plurality of coefficient sequences, to produce fifth and sixth pluralities of coefficient sequences;

and wherein the step of applying the reconstruction scaling and wavelet functions in a second direction comprise:

convolving the reconstruction scaling function with the fifth plurality of coefficient sequences, in the second direction;

convolving the reconstruction wavelet function with the sixth plurality of coefficient sequences, in the second direction;

after the convolving steps, summing each of the fifth plurality of coefficient sequences to an associated one of the sixth plurality of coefficient sequences, to produce a low-frequency component and a high-frequency component of the document image;

after the summing step, convolving the reconstruction scaling function with the low-frequency component of the document image, in the second direction;

convolving the reconstruction wavelet function with the high-frequency component of the document image, in the second direction; and

then summing the low-frequency and high-frequency components to produce an array corresponding to the document image.

22. A system for communicating image information corresponding to documents, comprising:

an input source for providing digital image information corresponding to a document to be communicated; and

compressor circuitry having an input coupled to said input source, for decomposing each frame of digital image information in a first image direction, by performing integer convolutions of the digital image information using a scaling function and a corresponding wavelet function that do not require support outside of the convolution interval, into a first low-frequency component and a first high-frequency component, and for then further decomposing each of said low-frequency and high-frequency components of each frame of digital video image information in a second image direction, by performing integer convolutions of the first low-frequency and high-frequency components using the scaling function and the corresponding wavelet function, into first and second pairs of low-frequency and second high-frequency components, said first pair being the decomposed representation of said first low-frequency component and said second pair corresponding to a decomposed representation of said first high-frequency component, said compressor circuitry also having an output for presenting the first and second pairs of low-frequency and high-frequency components.

23. The system of claim 22, further comprising:

a format converter circuit coupled between said input source and said compressor, for converting the digital input video information into a portable gray-level format.

24. The system of claim 23, wherein the format converter circuit comprises circuitry for byte-packing data corresponding to image data received from the input source.

25. The system of claim 24, wherein said input source provides color document information;

and wherein said byte-packing circuitry performs byte-packing of data containing more than one bit per picture element of the document.

26. The system of claim 22, wherein said compressor circuitry comprises:

decomposing circuitry for performing the decomposing of each frame of digital image information into the first and second pairs of low-frequency and high-frequency components;

quantization circuitry for quantizing the high-frequency component of said first pair, and for quantizing the both the low-frequency and the high-frequency component of the second pair; and

lossless compression circuitry, having an input coupled to said quantization circuitry, for performing lossless compression of the first and second low-frequency and high-frequency pairs, prior to presenting the decomposed frames of digital video image information at the output of the compressor circuitry.

27. The system of claim 26, further comprising:

main controller circuitry, for controlling the operation of the decomposing circuitry, so that the low-

frequency component of said first pair may be repetitively provided to the decomposing circuitry for further decomposing.

28. The system of claim 22, wherein said compressor circuitry comprises:

an image buffer coupled to the input of the compressor circuitry for storing a frame of digital video image information; and

a digital processor for performing the decomposing of each frame of digital image information into the first and second pairs of low-frequency and high-frequency components by performing integer convolutions of the digital image information using a scaling function and a corresponding wavelet function that do not require support outside of the convolution interval.

29. The system of claim 28, wherein said digital processor operates according to matrix operations using precalculated matrices corresponding to spline and wavelet function coefficient sequences stored in said compressor circuitry.

30. The system of claim 19, further comprising:

a communications network coupled to the output of the compressor circuitry;

decompressor circuitry, having an input coupled to said communications network, for reconstructing the decomposed digital image information in the second image direction by performing integer convolutions of the digital image information using a reconstruction scaling function and a corresponding reconstruction wavelet function that do not require support outside of the convolution interval, for then further reconstructing the digital image information in the first image direction by

performing integer convolutions of the digital image information using the reconstruction scaling and wavelet functions; and

means for displaying the reconstructed frames of digital image information.

31. A system for displaying compressed image data corresponding to a document, comprising:

a memory for storing the compressed image data;

decompressor circuitry for reconstructing the compressed image data in a first image direction by performing integer convolutions of the digital image information using a reconstruction scaling function and a corresponding reconstruction wavelet function that do not require support outside of the convolution interval, and for then further reconstructing the compressed image data in a second image direction by performing integer convolutions of the digital image information using the reconstruction scaling and wavelet functions; and

means, coupled to said decompressor circuitry, for outputting the reconstructed compressed image data.

32. The system of claim 31, wherein said memory is coupled to a communications network.

33. The system of claim 31, wherein said decompressor circuitry comprises:

a digital processor coupled to said memory, for reconstructing the digital image information using precalculated matrices of integers.

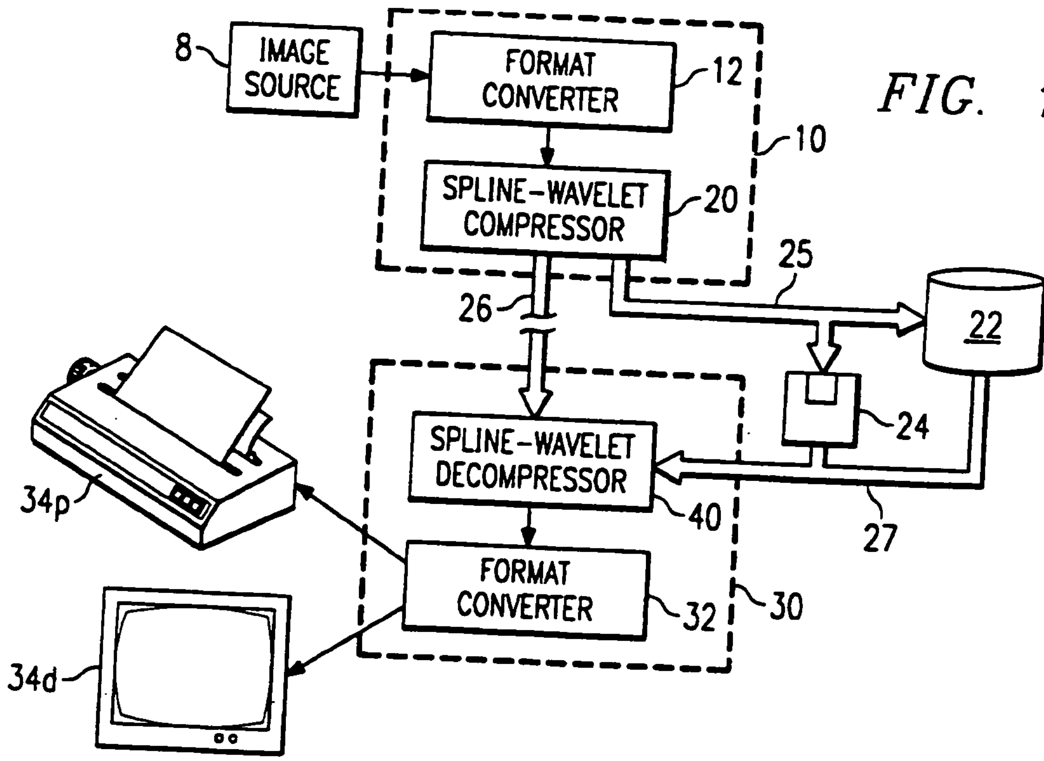


IMAGE SOURCE
8
FORMAT CONVERTER
12
10
SPLINE-WAVELET COMPRESSOR
20
25
26
22
24
SPLINE-WAVELET DECOMPRESSOR
40
27
34p
FORMAT CONVERTER
32
30
34d

FIG. 1

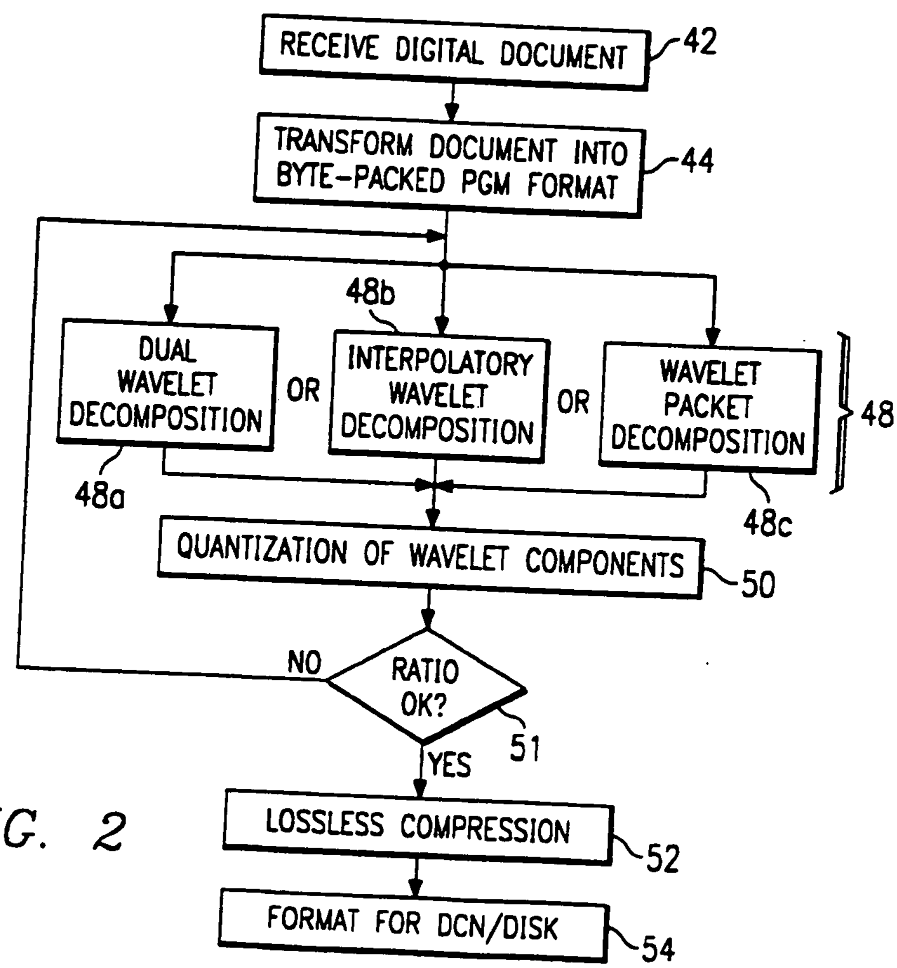

RECEIVE DIGITAL DOCUMENT
42
TRANSFORM DOCUMENT INTO BYTE-PACKED PGM FORMAT
44
48b
DUAL WAVELET DECOMPOSITION
OR
INTERPOLATORY WAVELET DECOMPOSITION
OR
WAVELET PACKET DECOMPOSITION
48
48a
48c
QUANTIZATION OF WAVELET COMPONENTS
50
NO
RATIO OK?
51
YES
LOSSLESS COMPRESSION
52
FORMAT FOR DCN/DISK
54

FIG. 2

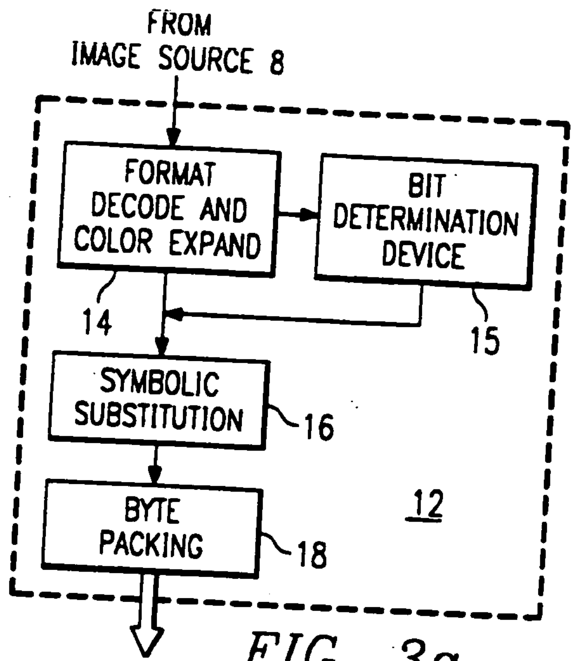


FIG. 3a

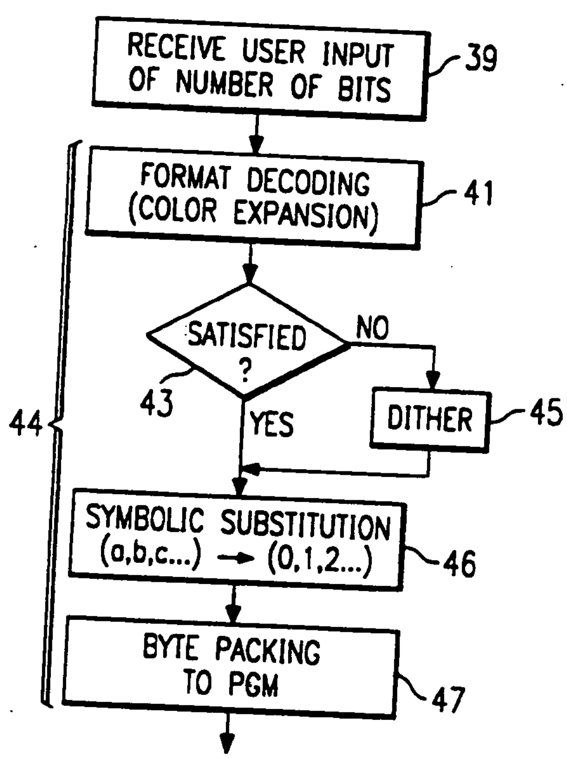


FIG. 3b

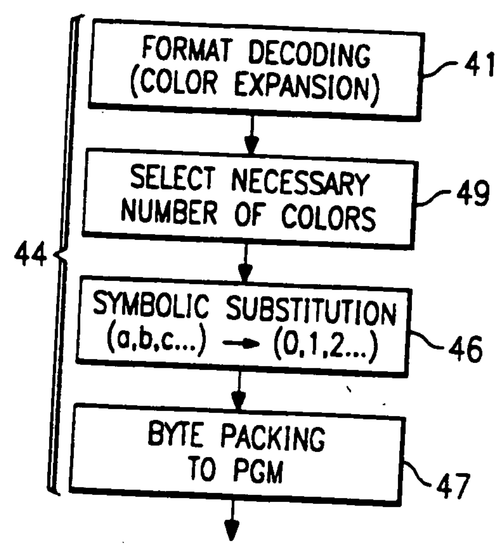


FIG. 3c

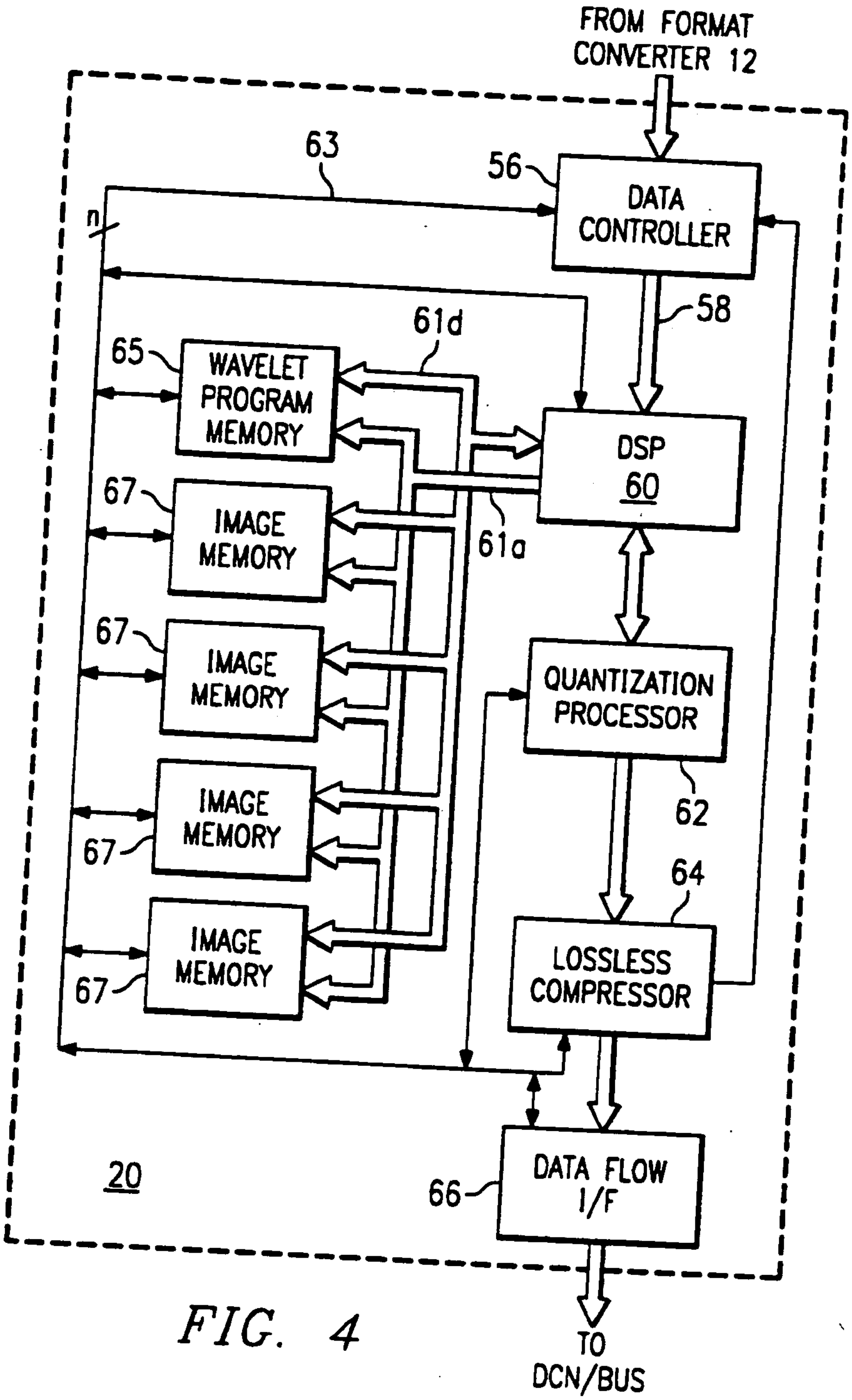

FROM FORMAT
CONVERTER 12
63
56
DATA
CONTROLLER
n
58
61d
65
WAVELET
PROGRAM
MEMORY
DSP
60
67
IMAGE
MEMORY
61a
67
IMAGE
MEMORY
QUANTIZATION
PROCESSOR
62
IMAGE
MEMORY
67
64
IMAGE
MEMORY
67
LOSSLESS
COMPRESSOR
20
66
DATA FLOW
I/F
TO
DCN/BUS

FIG. 4

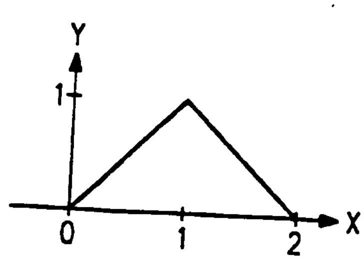

Y
1
0
1
2
X

FIG. 6

FIG. 5a

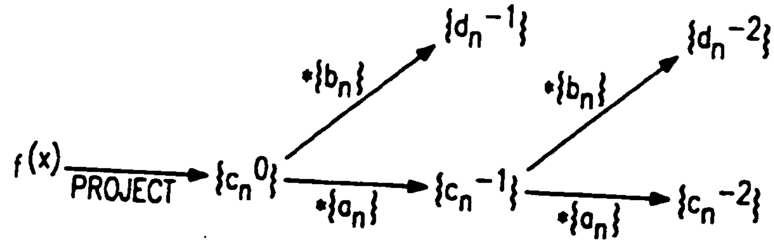
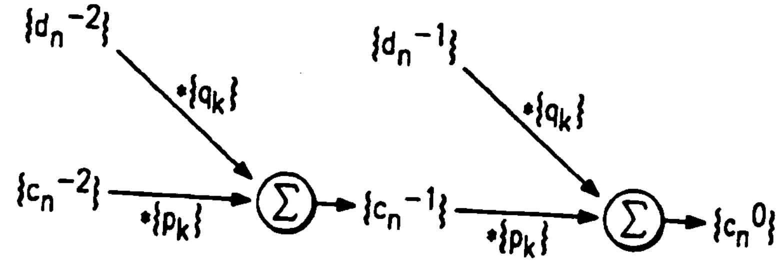


FIG. 5b



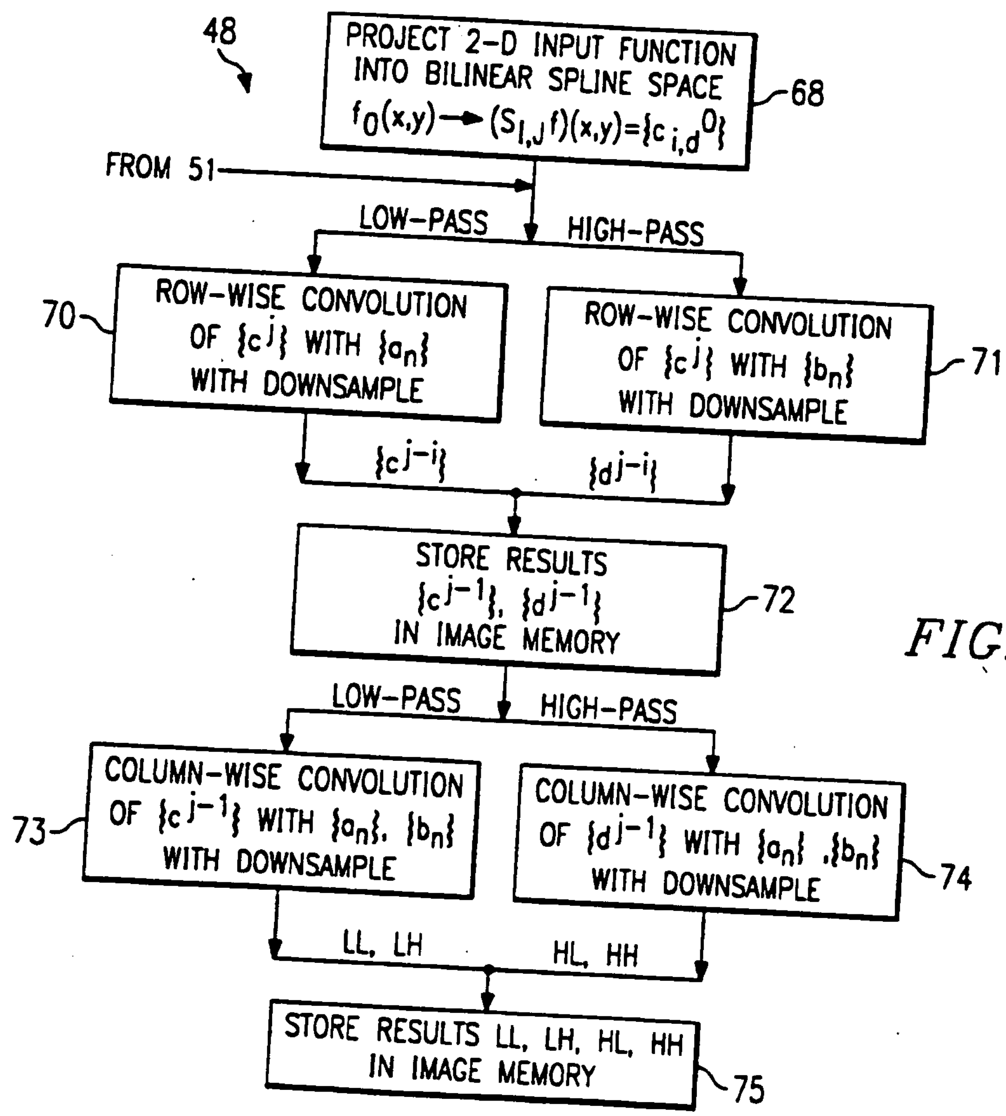


FIG. 7

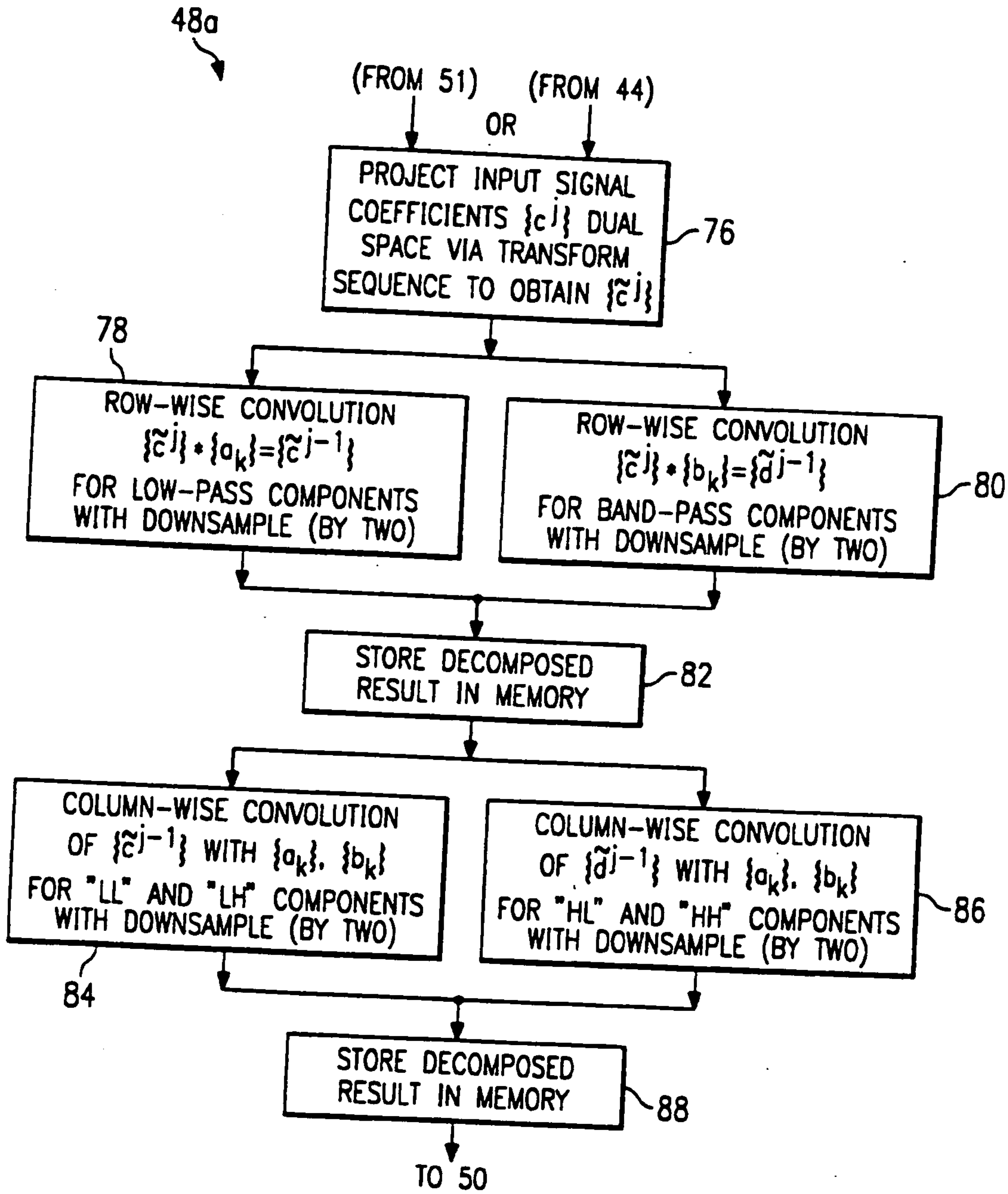

48a
(FROM 51)
OR
(FROM 44)
PROJECT INPUT SIGNAL COEFFICIENTS {c^j} DUAL SPACE VIA TRANSFORM SEQUENCE TO OBTAIN {c̃^j}
76
78
ROW-WISE CONVOLUTION {c̃^j}*{a_k}={c̃^(j-1)} FOR LOW-PASS COMPONENTS WITH DOWNSAMPLE (BY TWO)
ROW-WISE CONVOLUTION {c̃^j}*{b_k}={d̃^(j-1)} FOR BAND-PASS COMPONENTS WITH DOWNSAMPLE (BY TWO)
80
STORE DECOMPOSED RESULT IN MEMORY
82
COLUMN-WISE CONVOLUTION OF {c̃^(j-1)} WITH {a_k}, {b_k} FOR "LL" AND "LH" COMPONENTS WITH DOWNSAMPLE (BY TWO)
84
COLUMN-WISE CONVOLUTION OF {d̃^(j-1)} WITH {a_k}, {b_k} FOR "HL" AND "HH" COMPONENTS WITH DOWNSAMPLE (BY TWO)
86
STORE DECOMPOSED RESULT IN MEMORY
88
TO 50

FIG. 8

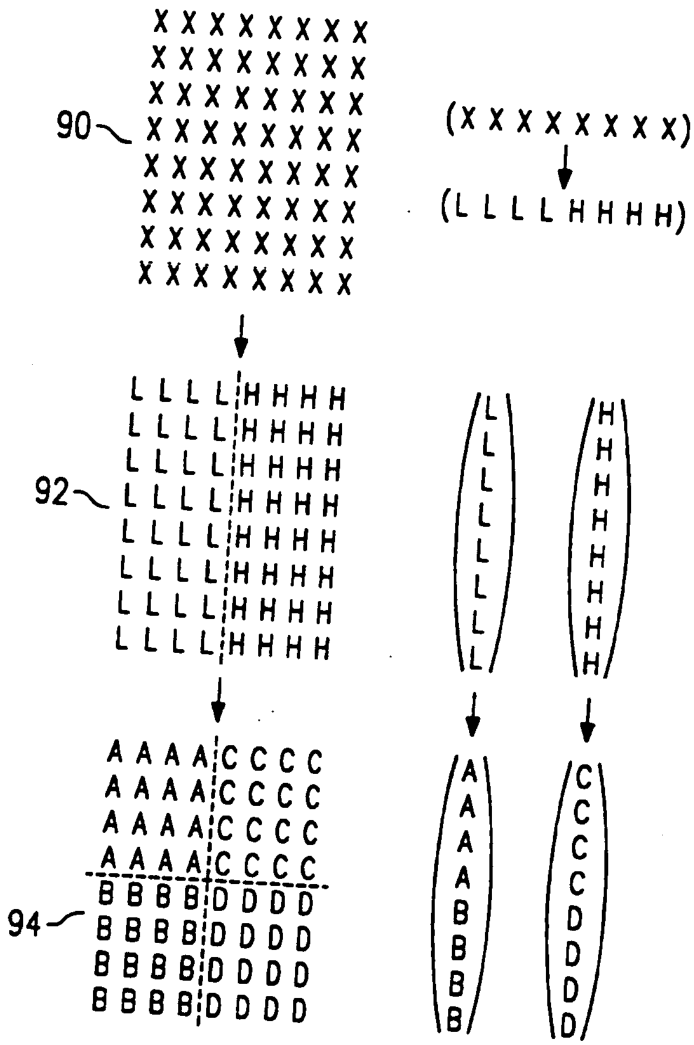


FIG. 9

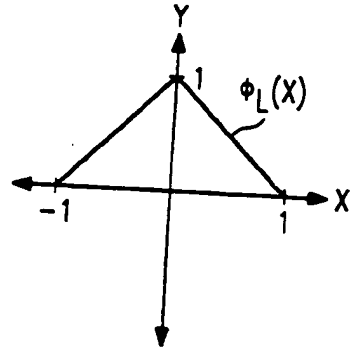


FIG. 10a

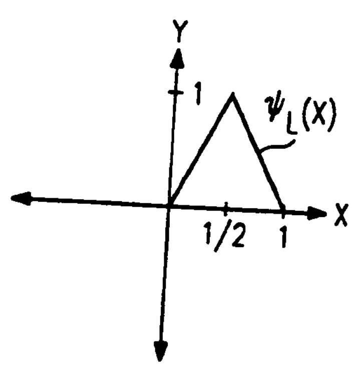


FIG. 10b

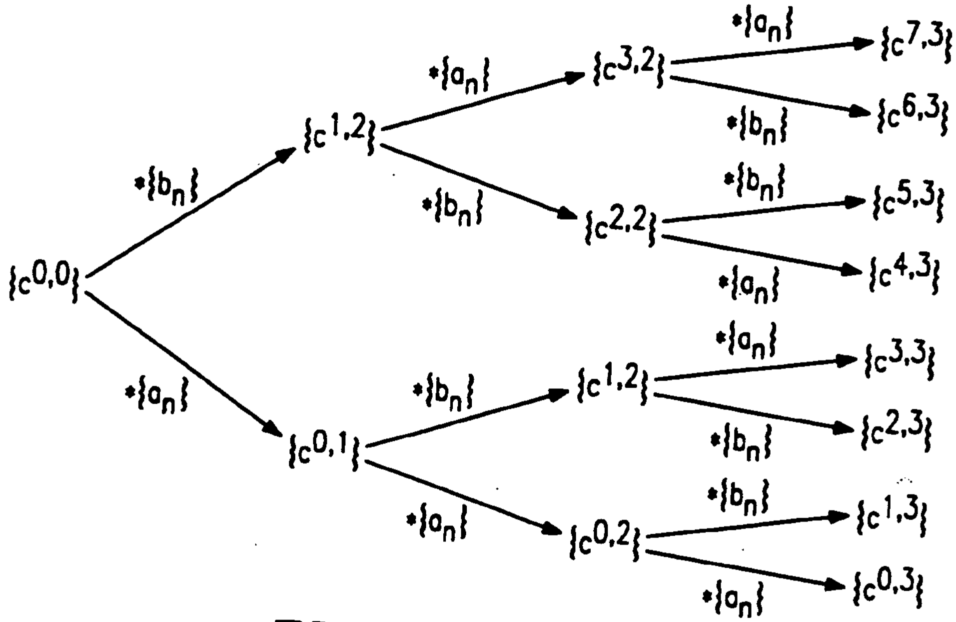


FIG. 11a

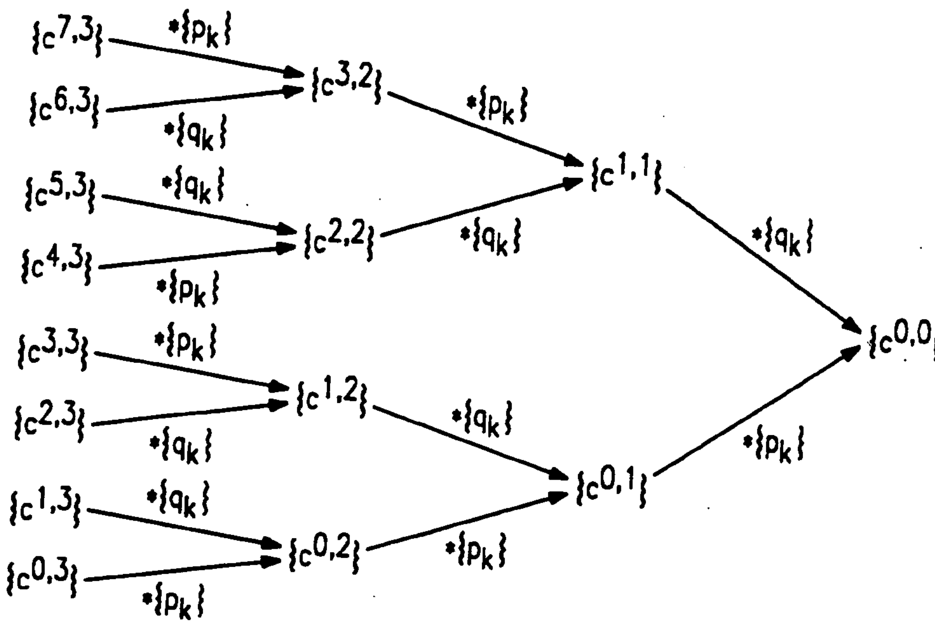


FIG. 11b

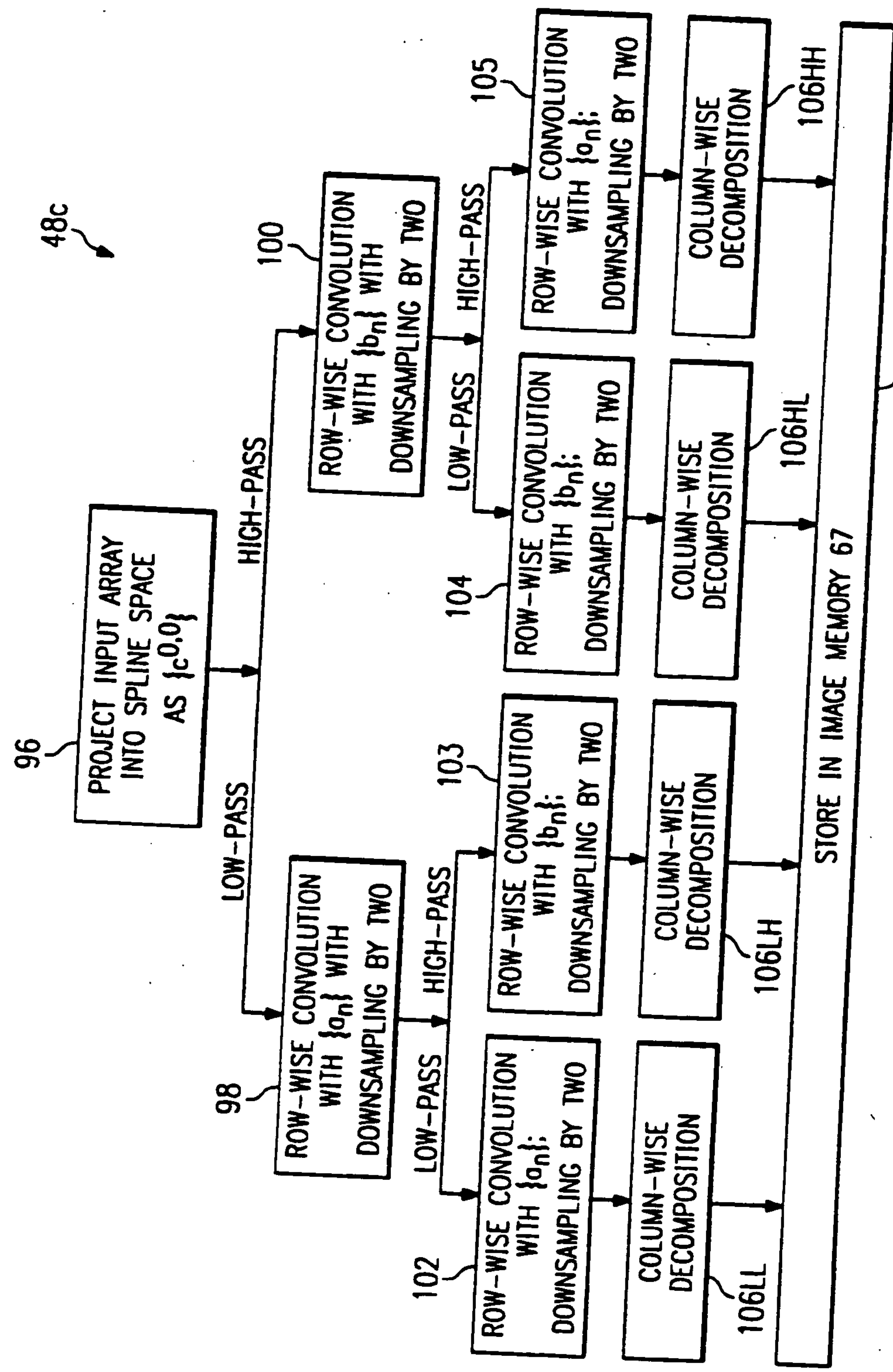

48c
96
PROJECT INPUT ARRAY INTO SPLINE SPACE AS $\{c^{0,0}\}$
LOW-PASS
HIGH-PASS
98
ROW-WISE CONVOLUTION WITH $\{a_n\}$ WITH DOWNSAMPLING BY TWO
100
ROW-WISE CONVOLUTION WITH $\{b_n\}$ WITH DOWNSAMPLING BY TWO
LOW-PASS
HIGH-PASS
LOW-PASS
HIGH-PASS
102
ROW-WISE CONVOLUTION WITH $\{a_n\}$: DOWNSAMPLING BY TWO
103
ROW-WISE CONVOLUTION WITH $\{b_n\}$: DOWNSAMPLING BY TWO
104
ROW-WISE CONVOLUTION WITH $\{b_n\}$: DOWNSAMPLING BY TWO
105
ROW-WISE CONVOLUTION WITH $\{a_n\}$: DOWNSAMPLING BY TWO
COLUMN-WISE DECOMPOSITION
106LL
COLUMN-WISE DECOMPOSITION
106LH
COLUMN-WISE DECOMPOSITION
106HL
COLUMN-WISE DECOMPOSITION
106HH
STORE IN IMAGE MEMORY 67
116

FIG. 12a

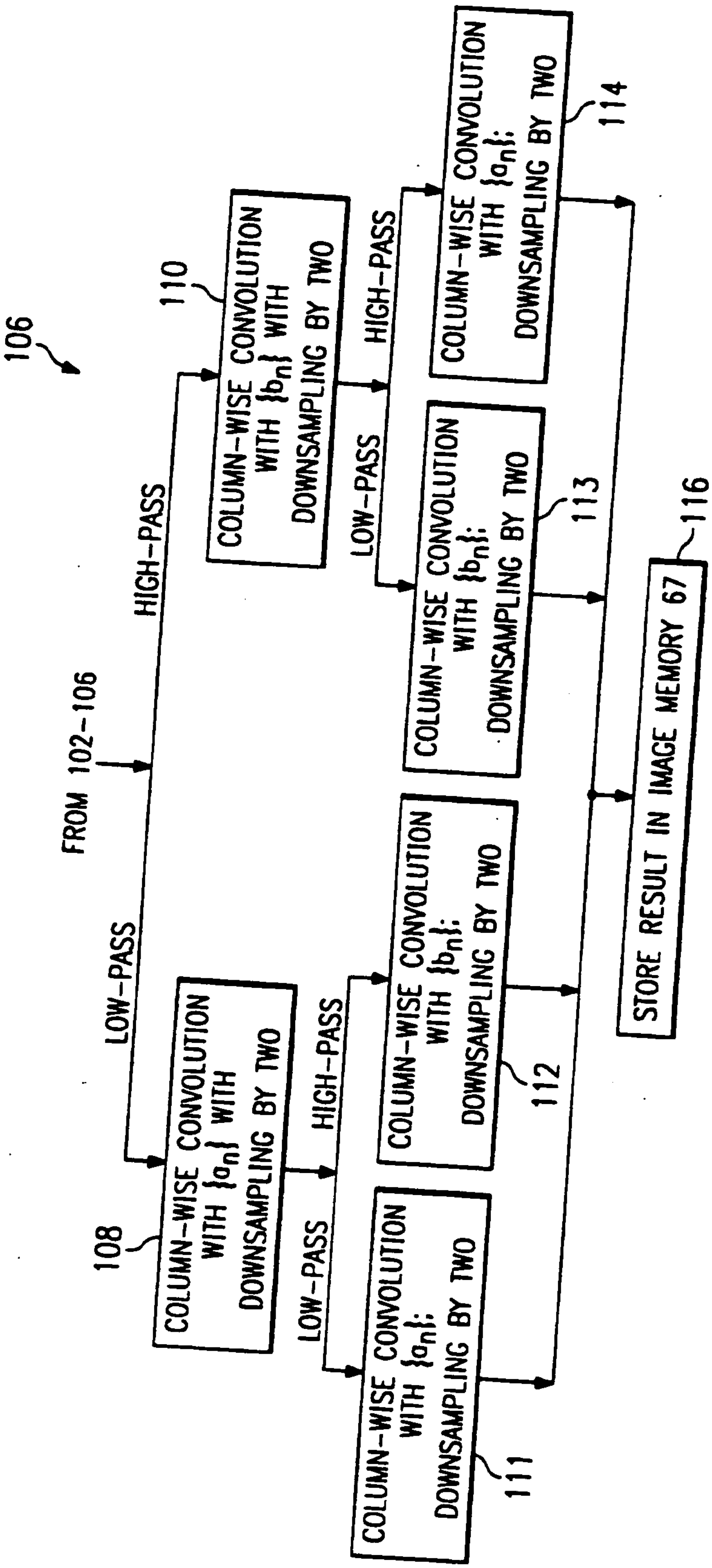

106
FROM 102-106
LOW-PASS
HIGH-PASS
COLUMN-WISE CONVOLUTION WITH {a_n} WITH DOWNSAMPLING BY TWO
108
COLUMN-WISE CONVOLUTION WITH {b_n} WITH DOWNSAMPLING BY TWO
110
LOW-PASS
HIGH-PASS
LOW-PASS
HIGH-PASS
COLUMN-WISE CONVOLUTION WITH {a_n}: DOWNSAMPLING BY TWO
111
COLUMN-WISE CONVOLUTION WITH {b_n}: DOWNSAMPLING BY TWO
112
COLUMN-WISE CONVOLUTION WITH {b_n}: DOWNSAMPLING BY TWO
113
COLUMN-WISE CONVOLUTION WITH {a_n}: DOWNSAMPLING BY TWO
114
STORE RESULT IN IMAGE MEMORY 67
116

FIG. 12b

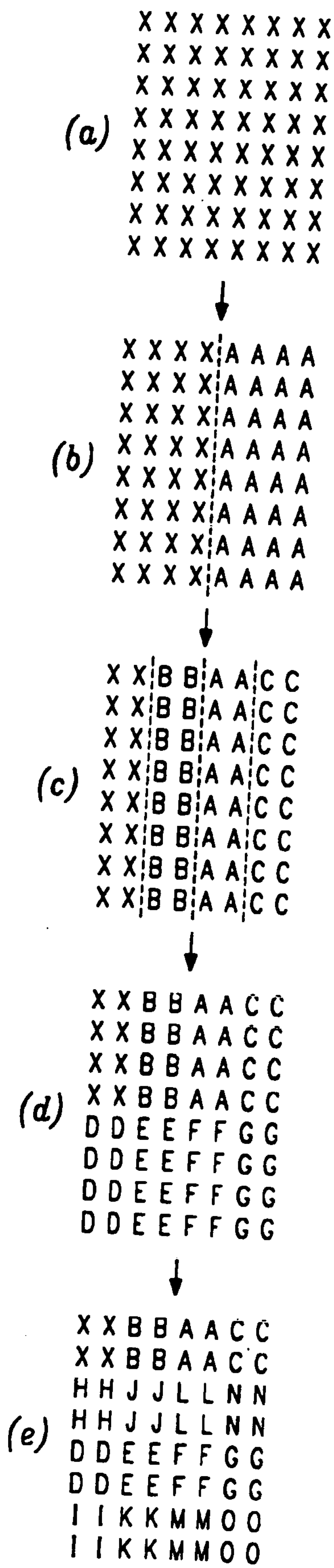

(a)
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXXXXXX
(b)
XXXXAAAA
XXXXAAAA
XXXXAAAA
XXXXAAAA
XXXXAAAA
XXXXAAAA
XXXXAAAA
XXXXAAAA
(c)
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
(d)
XXBBAACC
XXBBAACC
XXBBAACC
XXBBAACC
DDEEFFGG
DDEEFFGG
DDEEFFGG
DDEEFFGG
(e)
XXBBAACC
XXBBAACC
HHJJLLNN
HHJJLLNN
DDEEFFGG
DDEEFFGG
IIKKMMOO
IIKKMMOO

FIG. 13

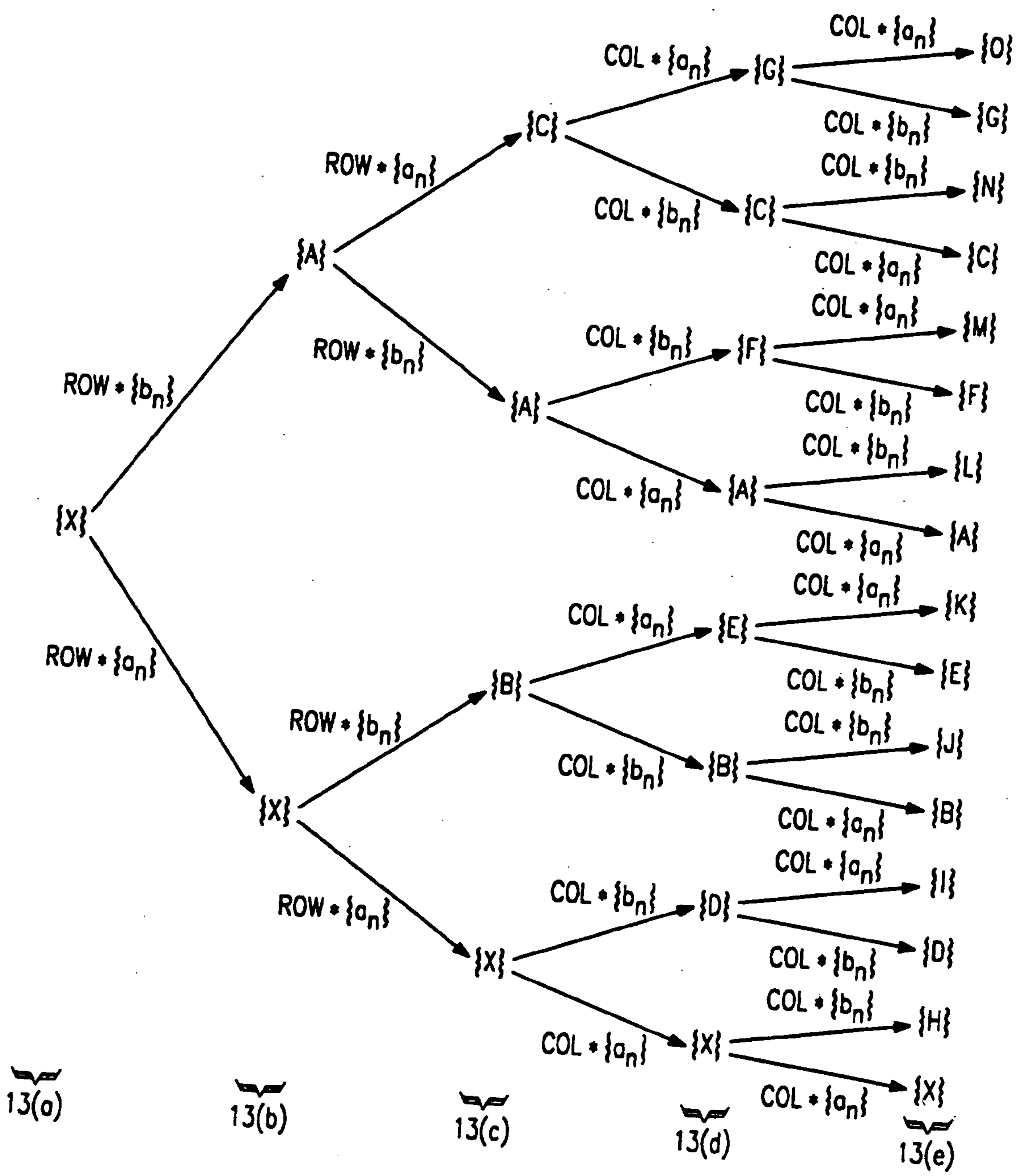

{X}
ROW * {b_n}
ROW * {a_n}
{A}
ROW * {a_n}
ROW * {b_n}
{C}
{A}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
{G}
{C}
{F}
{A}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
{O}
{G}
{N}
{C}
{M}
{F}
{L}
{A}
{X}
ROW * {b_n}
ROW * {a_n}
{B}
{X}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
{E}
{B}
{D}
{X}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
COL * {a_n}
COL * {b_n}
COL * {b_n}
COL * {a_n}
{K}
{E}
{J}
{B}
{I}
{D}
{H}
{X}
13(a)
13(b)
13(c)
13(d)
13(e)

FIG. 14

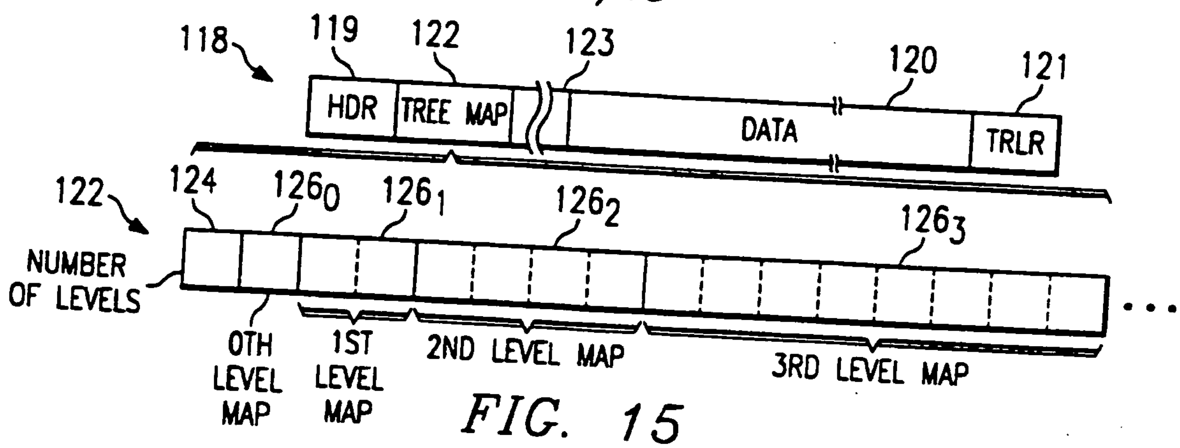
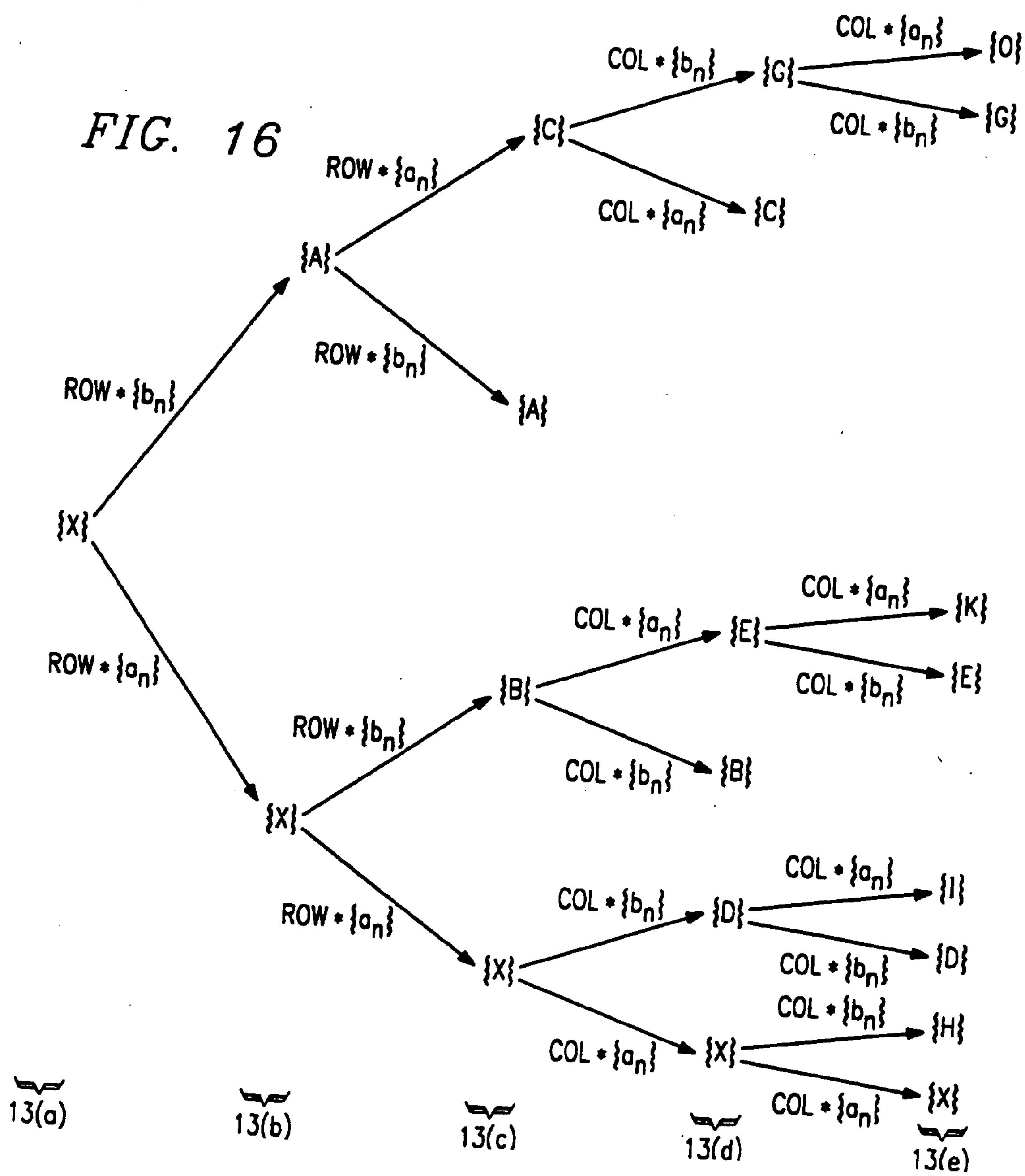

118
119
122
123
120
121
HDR
TREE MAP
DATA
TRLR
122
124
126_0
126_1
126_2
126_3
NUMBER OF LEVELS
0TH LEVEL MAP
1ST LEVEL MAP
2ND LEVEL MAP
3RD LEVEL MAP

FIG. 15

FIG. 16


{X}
ROW * $\{b_n\}$
ROW * $\{a_n\}$
{A}
ROW * $\{a_n\}$
ROW * $\{b_n\}$
{C}
{A}
COL * $\{b_n\}$
COL * $\{a_n\}$
{G}
{C}
COL * $\{a_n\}$
COL * $\{b_n\}$
{O}
{G}
{X}
ROW * $\{b_n\}$
ROW * $\{a_n\}$
{B}
COL * $\{a_n\}$
COL * $\{b_n\}$
{E}
{B}
COL * $\{a_n\}$
COL * $\{b_n\}$
{K}
{E}
{X}
COL * $\{b_n\}$
COL * $\{a_n\}$
{D}
COL * $\{a_n\}$
COL * $\{b_n\}$
{I}
{D}
{X}
COL * $\{b_n\}$
COL * $\{a_n\}$
{H}
{X}
13(a)
13(b)
13(c)
13(d)
13(e)

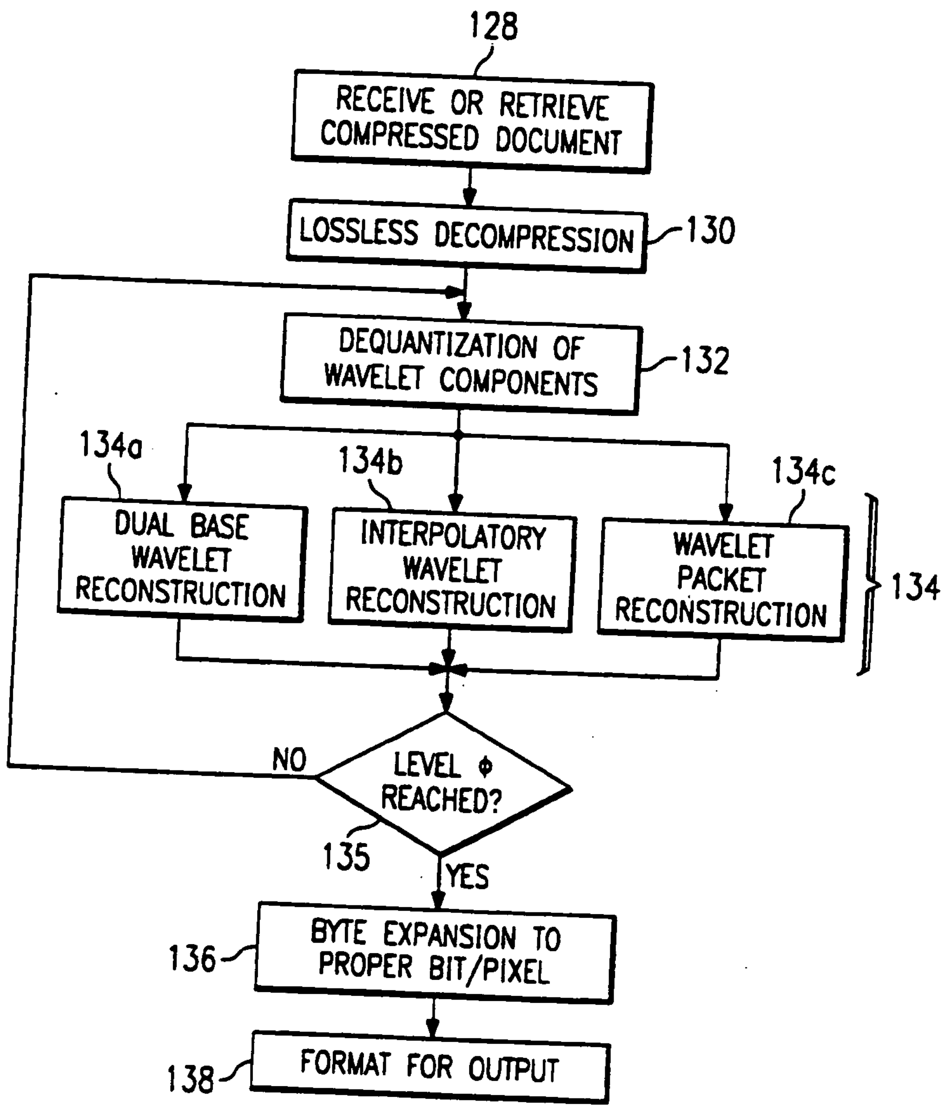

128
RECEIVE OR RETRIEVE COMPRESSED DOCUMENT
LOSSLESS DECOMPRESSION
130
DEQUANTIZATION OF WAVELET COMPONENTS
132
134a
134b
134c
DUAL BASE WAVELET RECONSTRUCTION
INTERPOLATORY WAVELET RECONSTRUCTION
WAVELET PACKET RECONSTRUCTION
134
NO
LEVEL ϕ REACHED?
135
YES
136
BYTE EXPANSION TO PROPER BIT/PIXEL
138
FORMAT FOR OUTPUT

FIG. 17

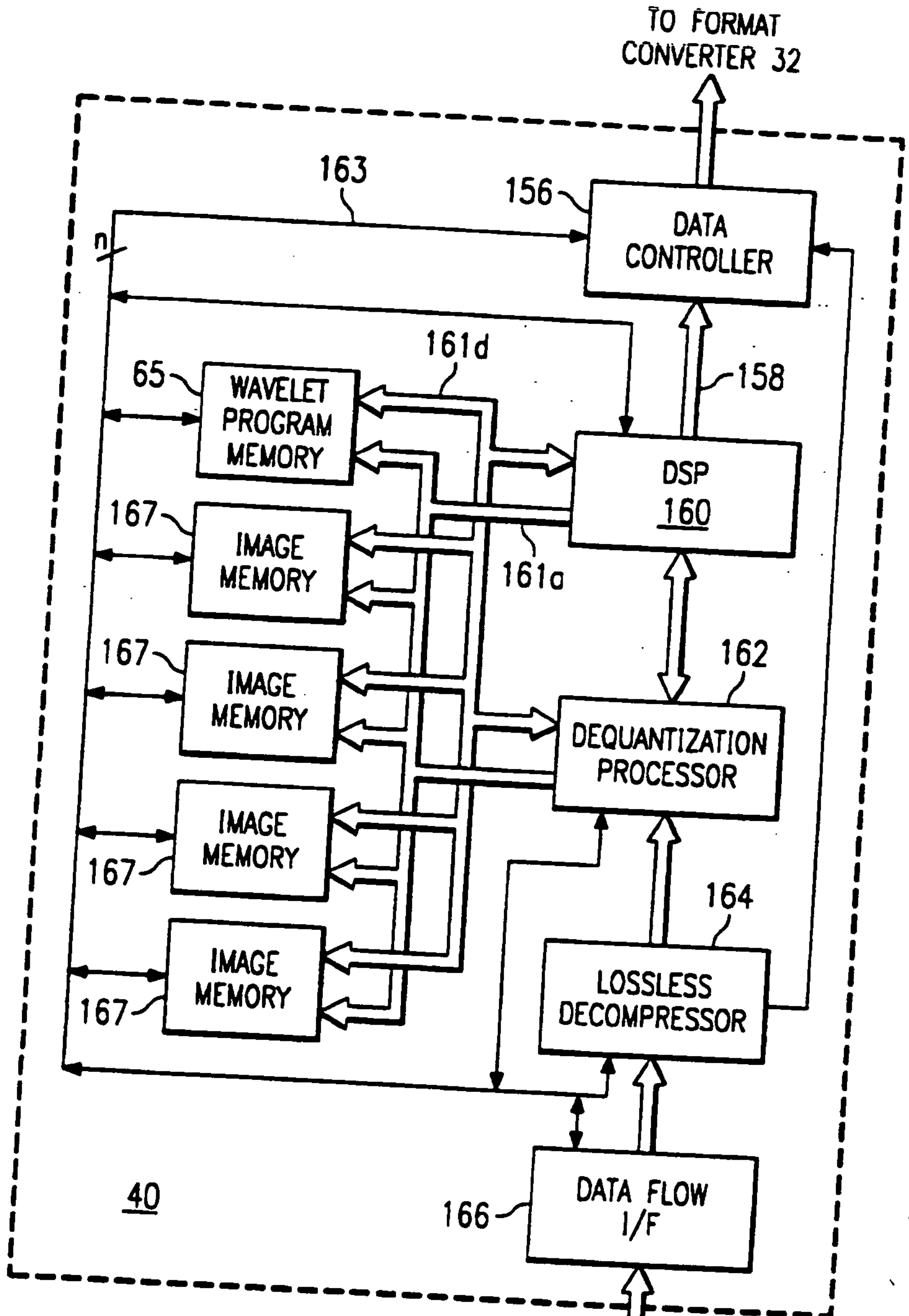

TO FORMAT CONVERTER 32
163
156
DATA CONTROLLER
n
161d
158
65
WAVELET PROGRAM MEMORY
DSP 160
167
IMAGE MEMORY
161a
167
IMAGE MEMORY
162
DEQUANTIZATION PROCESSOR
IMAGE MEMORY
167
164
IMAGE MEMORY
167
LOSSLESS DECOMPRESSOR
40
DATA FLOW I/F
166
FROM DCN/BUS

FIG. 18

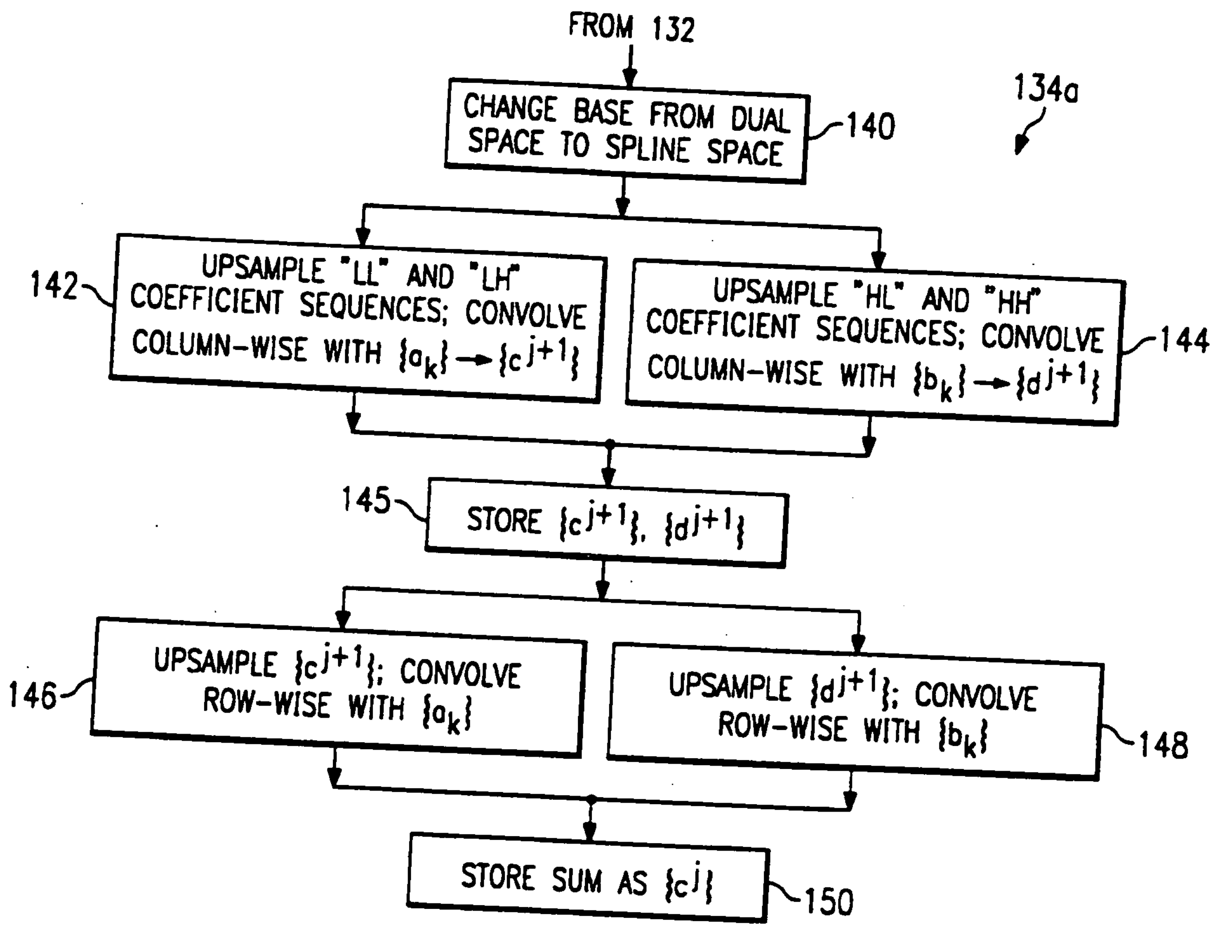

FROM 132
134a
CHANGE BASE FROM DUAL SPACE TO SPLINE SPACE
140
142
UPSAMPLE "LL" AND "LH" COEFFICIENT SEQUENCES; CONVOLVE COLUMN-WISE WITH $\{a_k\} \rightarrow \{c^{j+1}\}$
UPSAMPLE "HL" AND "HH" COEFFICIENT SEQUENCES; CONVOLVE COLUMN-WISE WITH $\{b_k\} \rightarrow \{d^{j+1}\}$
144
145
STORE $\{c^{j+1}\}$, $\{d^{j+1}\}$
146
UPSAMPLE $\{c^{j+1}\}$; CONVOLVE ROW-WISE WITH $\{a_k\}$
UPSAMPLE $\{d^{j+1}\}$; CONVOLVE ROW-WISE WITH $\{b_k\}$
148
STORE SUM AS $\{c^j\}$
150

FIG. 19

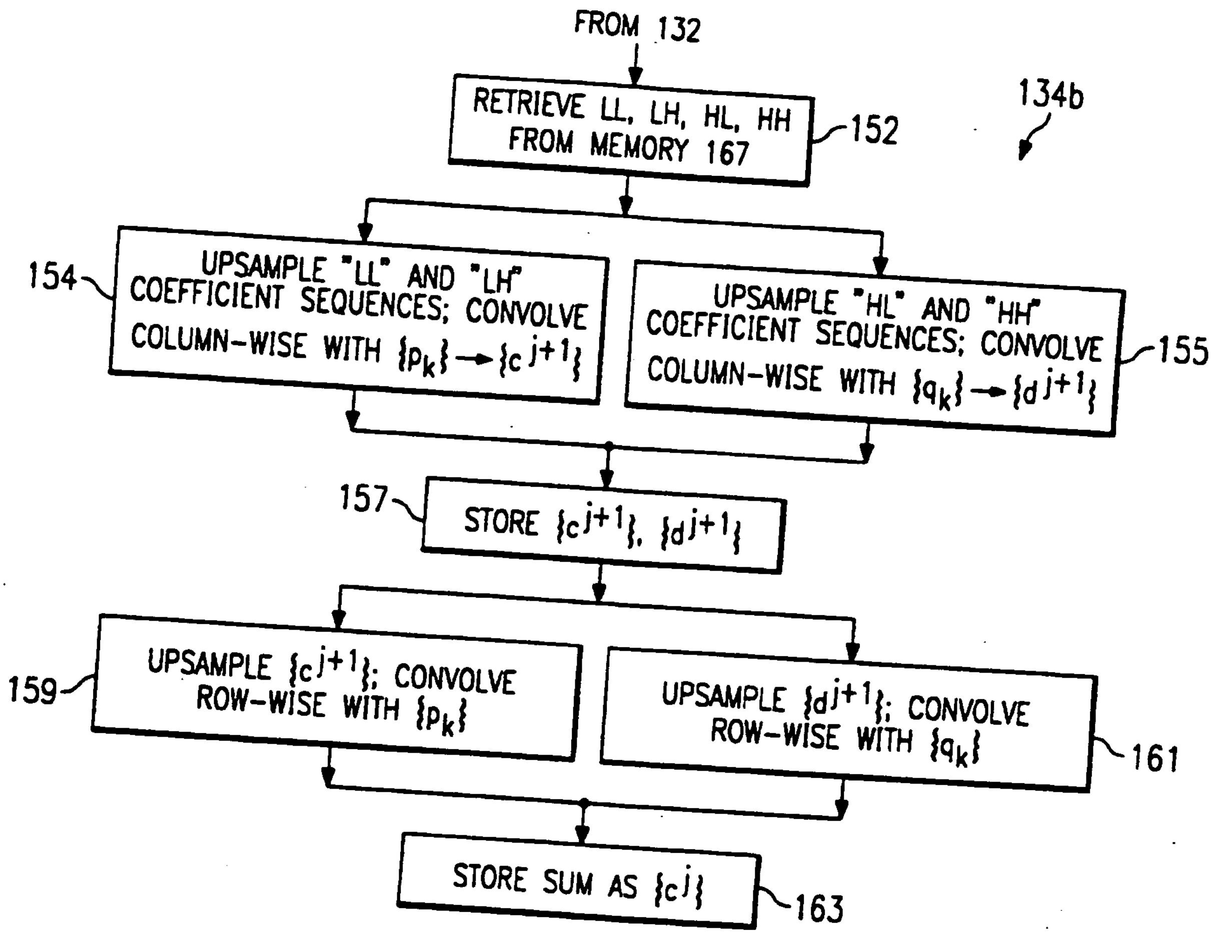

FROM 132
RETRIEVE LL, LH, HL, HH FROM MEMORY 167
152
134b
154
UPSAMPLE "LL" AND "LH" COEFFICIENT SEQUENCES; CONVOLVE COLUMN-WISE WITH {p_k} → {c^{j+1}}
UPSAMPLE "HL" AND "HH" COEFFICIENT SEQUENCES; CONVOLVE COLUMN-WISE WITH {q_k} → {d^{j+1}}
155
157
STORE {c^{j+1}}, {d^{j+1}}
159
UPSAMPLE {c^{j+1}}; CONVOLVE ROW-WISE WITH {p_k}
UPSAMPLE {d^{j+1}}; CONVOLVE ROW-WISE WITH {q_k}
161
STORE SUM AS {c^j}
163

FIG. 20

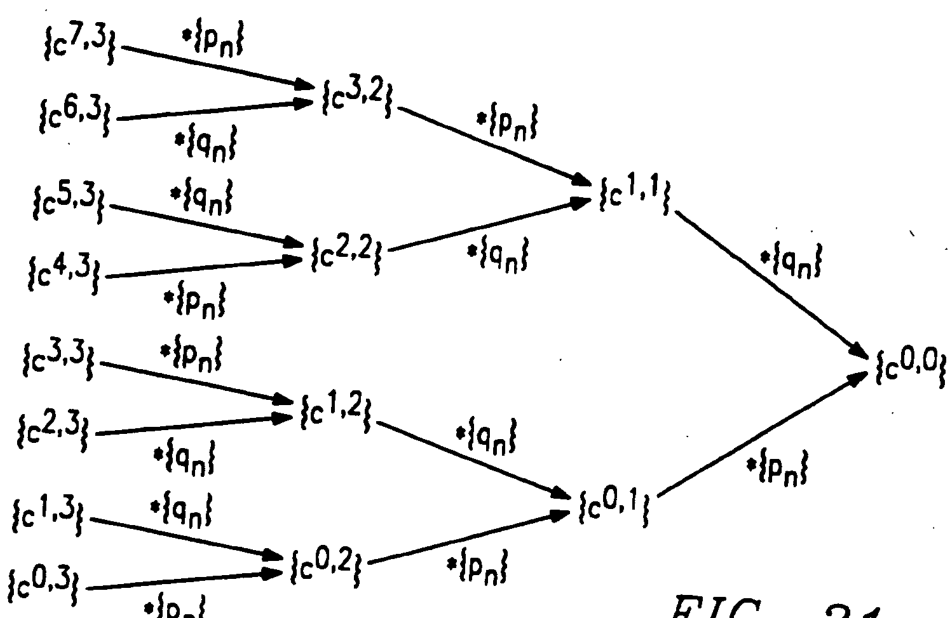

{c^{7,3}}
*{p_n}
{c^{6,3}}
*{q_n}
{c^{3,2}}
*{p_n}
{c^{5,3}}
*{q_n}
{c^{4,3}}
*{p_n}
{c^{2,2}}
*{q_n}
{c^{1,1}}
*{q_n}
{c^{3,3}}
*{p_n}
{c^{2,3}}
*{q_n}
{c^{1,2}}
*{q_n}
{c^{1,3}}
*{q_n}
{c^{0,3}}
*{p_n}
{c^{0,2}}
*{p_n}
{c^{0,1}}
*{p_n}
{c^{0,0}}

FIG. 21

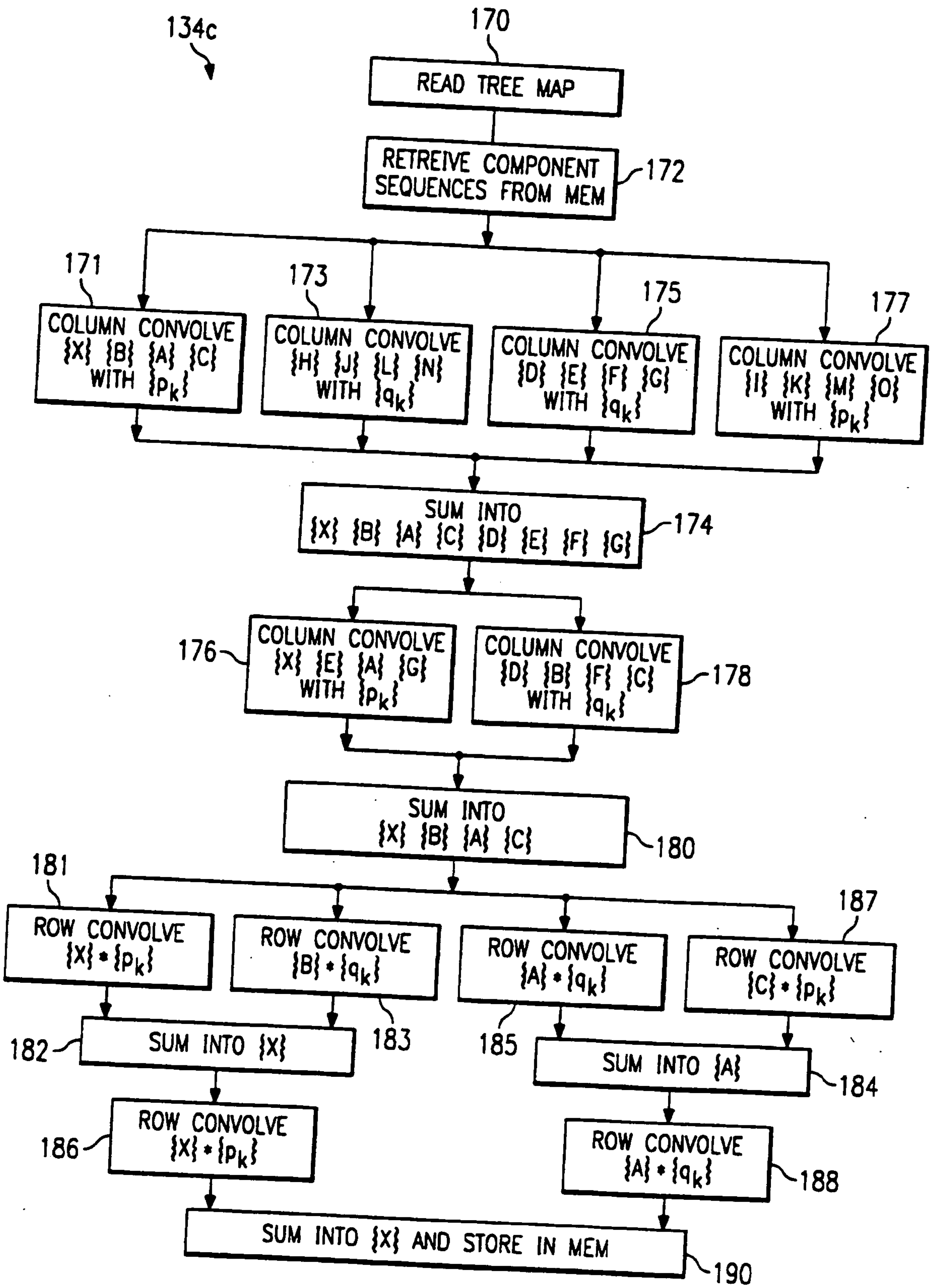

134c
170
READ TREE MAP
RETREIVE COMPONENT SEQUENCES FROM MEM
172
171
173
175
177
COLUMN CONVOLVE {X} {B} {A} {C} WITH {p_k}
COLUMN CONVOLVE {H} {J} {L} {N} WITH {q_k}
COLUMN CONVOLVE {D} {E} {F} {G} WITH {q_k}
COLUMN CONVOLVE {I} {K} {M} {O} WITH {p_k}
SUM INTO {X} {B} {A} {C} {D} {E} {F} {G}
174
176
COLUMN CONVOLVE {X} {E} {A} {G} WITH {p_k}
COLUMN CONVOLVE {D} {B} {F} {C} WITH {q_k}
178
SUM INTO {X} {B} {A} {C}
180
181
187
ROW CONVOLVE {X} * {p_k}
ROW CONVOLVE {B} * {q_k}
ROW CONVOLVE {A} * {q_k}
ROW CONVOLVE {C} * {p_k}
182
SUM INTO {X}
183
185
SUM INTO {A}
184
186
ROW CONVOLVE {X} * {p_k}
ROW CONVOLVE {A} * {q_k}
188
SUM INTO {X} AND STORE IN MEM
190

FIG. 22

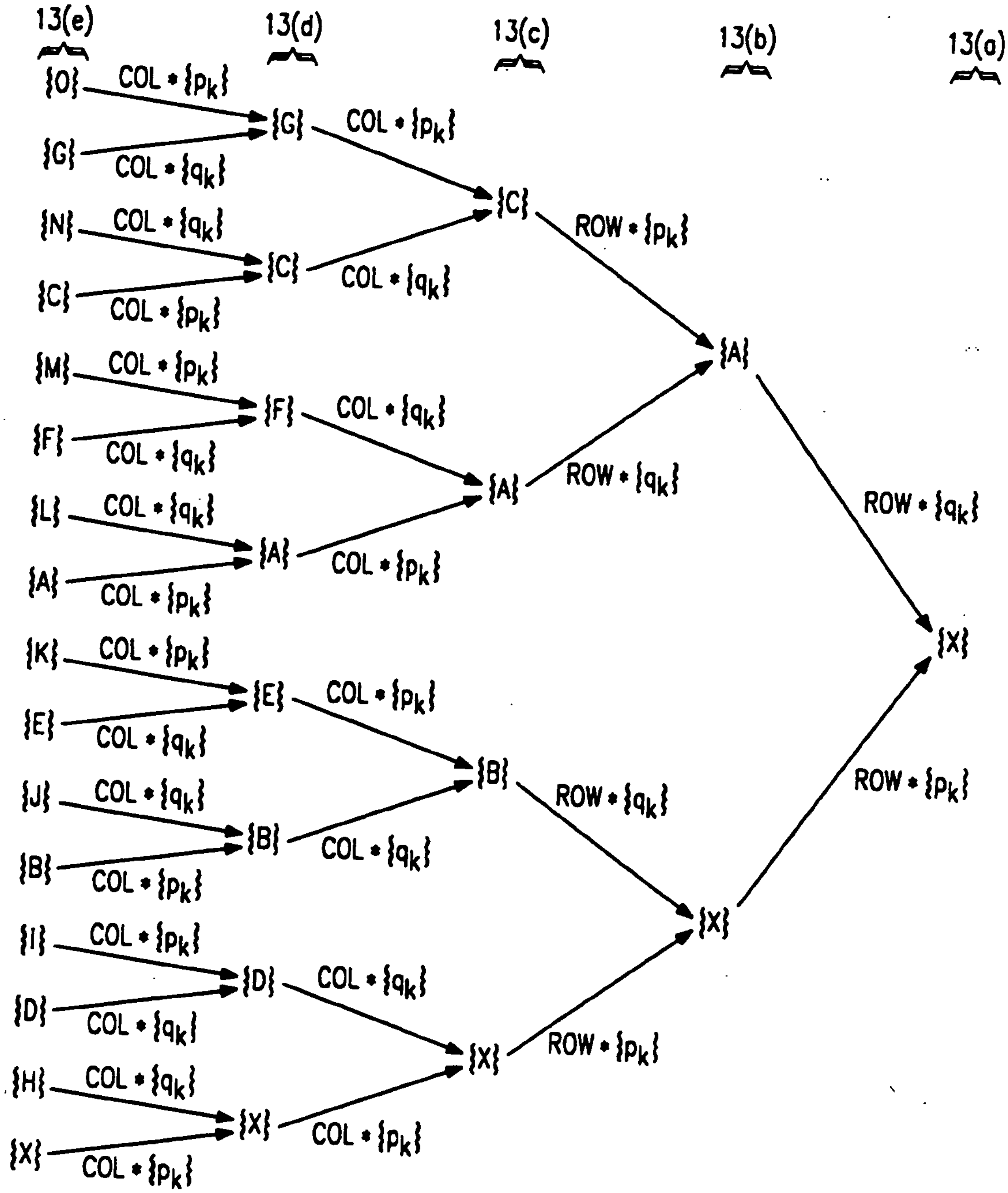

13(e)
13(d)
13(c)
13(b)
13(a)
{O}
COL * {p_k}
{G}
COL * {q_k}
{G}
COL * {p_k}
{C}
{N}
COL * {q_k}
{C}
COL * {p_k}
{C}
COL * {q_k}
ROW * {p_k}
{A}
{M}
COL * {p_k}
{F}
COL * {q_k}
{F}
COL * {q_k}
{A}
ROW * {q_k}
{L}
COL * {q_k}
{A}
COL * {p_k}
{A}
COL * {p_k}
ROW * {q_k}
{X}
{K}
COL * {p_k}
{E}
COL * {q_k}
{E}
COL * {p_k}
{B}
ROW * {q_k}
{J}
COL * {q_k}
{B}
COL * {p_k}
{B}
COL * {q_k}
ROW * {p_k}
{X}
{I}
COL * {p_k}
{D}
COL * {q_k}
{D}
COL * {q_k}
{X}
ROW * {p_k}
{H}
COL * {q_k}
{X}
COL * {p_k}
{X}
COL * {p_k}

FIG. 23

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/26, G06T 1/00 | A1

(11) International Publication Number: WO 95/19683

(43) International Publication Date: 20 July 1995 (20.07.95)

(21) International Application Number: PCT/US95/00563

(22) International Filing Date: 13 January 1995 (13.01.95)

(30) Priority Data:
08/181,663 14 January 1994 (14.01.94) US

(71) Applicant: HOUSTON ADVANCED RESEARCH CENTER [US/US]; 4802 Research Forest Drive, The Woodlands, TX 77381 (US).

(72) Inventors: CHUI, Charles, K.; 2120 Carter Lake Drive, College Station, TX 77842 (US). YUEN, Pak-Kay; 1528 Hillside Drive, College Station, TX 77840 (US).

(74) Agents: ANDERSON, Rodney, M. et al.; Vinson & Elkins L.L.P., 2500 First City Tower, 1001 Fannin Street, Houston, TX 77002 (US).

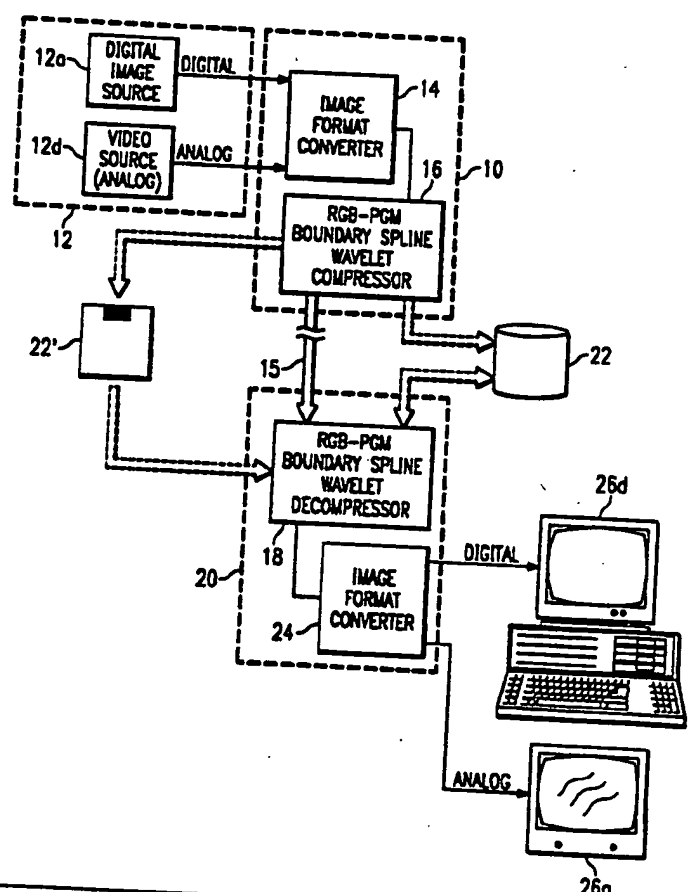
(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, MX, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SI, SK, TJ, TT, UA, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: BOUNDARY-SPLINE-WAVELET COMPRESSION FOR VIDEO IMAGES

(57) Abstract

A method and apparatus for performing video image compression and decompression are disclosed. The video image compression is performed using boundary-spline-wavelet decomposition, in which the wavelets applied to sample locations at the boundaries of image intervals are different from those applied to sample locations within the intervals. The decomposition is performed first for horizontal rows of the image data, and then in a vertical direction upon the results of the first decomposition. Quantization serves to locally round off the higher frequency components of the decomposition, and the decomposition is repeated until the desired compression ratio is obtained. Lossless compression may then be applied to the decomposed image data, and the compressed image is transmitted or stored, depending upon the application. Decompression is effected by lossless decompression of the received data, followed by reconstruction of the image using boundary-spline-wavelets, repeated as necessary to fully reconstruct the image. The reconstructed image can then be displayed on a conventional video display.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgystan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

BOUNDARY-SPLINE-WAVELET COMPRESSION FOR VIDEO IMAGES

* * * * *

This invention is in the field of video image storage and communication, and is more specifically directed to compression and decompression of video images, including motion video images and high-quality still video images.

Background of the Invention

1. Conventional video compression

Modern electronic data processing systems are now often used not only for conventional numeric and text processing tasks, but also for processing data corresponding to visual images. As such, many modern computer and telecommunications systems have "multimedia" capabilities, where the data processed and communicated includes video images either generated by a computer or digitized from a conventional video camera.

The capacity required to communicate video images on a real-time basis is huge, however, when measured against modern capabilities. For example, a single video image (i.e., frame) displayed by a rectangular array of picture elements ("pixels") arranged in 640 rows by 800 columns, with the color of each pixel represented by twenty-four digital bits, would require over 1.52 million bytes (1500 kbytes) of digital memory to store all information in the frame. While this memory requirement for a single video frame is large, digital storage of a series of frames, such as a motion picture, will quickly consume the disk storage capacity of even high-end computers and workstations.

The large amount of digital data necessary to represent a video frame not only impacts computer storage requirements, but limits the rate at which conventional systems can communicate motion pictures. Considering that conventional high-speed digital communication channels have a bandwidth of on the order of from 40 to 80 MHz, it becomes quite apparent that conventional motion pictures of thirty frames per second, with 1500 kbytes of digital information per frame, cannot be fully transmitted in real-time over state of the art digital communications systems.

In response to these limitations on the ability of modern computer systems to store and communicate video information, various types of data compression techniques have been developed in recent years. Conventional data compression techniques are generally referred to as of either "lossless" or "lossy", depending upon whether data is discarded in the compression process.

A survey of conventional lossless data compression is provided in Simon, "Lossless Compression: How it Works", PC Magazine (June 29, 1993), pp. 305-13. Examples of conventional lossless data compression techniques include Huffman encoding, Fano-Shannon encoding, and arithmetic encoding, as well as dynamic variations of the Huffman and Fano-Shannon probabilistic encoding techniques. In general, lossless compression techniques are primarily used to compress entered data such as data generated from data processing operations, rather than sampled data representative of analog video or audio signals, as decompression will reproduce all bits of the original compressed data stream.

Lossy data compression techniques, in contrast, provide additional data compression efficiency over

lossless data compression, as some amount of data is discarded in these techniques. As a result, lossy data compression techniques are generally used on sampled data, as some amount of inaccuracy relative to the true input data is tolerable; lossy data compression is, of course, inappropriate for use on entered data streams such as those produced by a data processing operation. Accordingly, lossy data compression techniques are widely used in the field of compression of video and motion picture images to obtain a high degree of compression, as some inaccuracy may be tolerated. A survey of conventional lossy data compression techniques may be found at Simon, "How Lossy Data Compression Shrinks Image Files", PC Magazine (July 1993), pp. 371 et seq.

A popular conventional lossy data compression technique is referred to as the JPEG (Joint Photographic Experts Group) method. A description of this technique may be found in Barnsley and Hurd, Fractal Image Compression (AK Peters, Ltd., 1993), pp. 219-228. The JPEG compression method initially divides the image into blocks of pixels, and a Discrete Cosine Transform (DCT) is performed on each pixel block, producing a representation of the block as coefficients corresponding to frequencies and amplitudes, rather than corresponding directly to color information. These coefficients are then quantized, or rounded off, and a difference algorithm is performed over all quantized blocks in the image, in a selected scan order. This difference algorithm subtracts a DC term corresponding to the mean pixel value of a block, from the DC term of the preceding block. The difference coefficients are then scanned in a different order, such as a zig-zag order, and the non-zero coefficients (i.e., blocks in which a difference from the preceding block occurred) are coded to indicate the number of preceding zero coefficients (i.e.,

the number of pixel blocks in which no change occurred) and also the value of the non-zero difference. Lossless compression is then often applied to the coded result to further compress the data. Decompression is performed by reversing the compression process, producing the displayable image.

While the JPEG conventional video image compression technique is useful in obtaining high degrees of compression, it has been found that JPEG compression is incapable of being used in a real-time fashion for a motion picture. This is because the time generally required to perform the JPEG decompression of a motion picture frame exceeds the display time for the frame (1/30 second), and as a result the motion picture image cannot be decompressed for real-time display. Temporally accurate display of a motion picture compressed according to these techniques, thus requires the decompression and display to be done in two steps, with the decompressed motion picture stored on video tape or another medium from which the motion picture can be played with the proper time base.

Another conventional method of lossy video image compression referred to as Recursive Vector Quantization (RVQ) quantizes the pixel blocks directly, without a DCT or other transform, according to a set of selected reference tiles. See Simon, July 1993, op. cit.. The reference tiles are selected according to an iterative technique, based upon the accuracy of the results relative to the original image. As noted in the Simon article, compression according to the RVQ method is computationally intense and complex, but decompression can be done quite rapidly.

Another type of conventional lossy video image compression techniques is referred to as fractal

compression. As is well known in the art, a fractal is a mathematical image object that is self-similar, in that the image can be represented in terms of other pieces of the image. In fractal image compression, the input image is similarly divided into pixel groups, or tiles. Each tile is then approximated by a transformation (contractive, rotational, or both) of one or more other reference regions of the image. The compressed image thus consists of a full representation of the reference region, plus the transformation operators for each of the tiles. Each tile of the image is decompressed by performing a transformation of the reference region using the stored transformation operator for that tile. Detailed descriptions of conventional fractal image compression techniques and systems for performing the same may be found in Barnsley & Hurd, Fractal Image Compression (AK Peters, Ltd., 1993), in U.S. Patent No. 4,941,193, and in U.S. Patent No. 5,065,447.

2. Frequency and time windowing functions

By way of further background, the field of wavelet analysis has recently become popular in the analysis of the time and frequency response and behavior of signals. The following section of this application is intended to provide a theoretical background for wavelet analysis techniques in order to both convey the state of the art in wavelet analysis, and also to provide the necessary background for the person of ordinary skill in the art to fully appreciate the present invention.

In the general sense, wavelet analysis is concerned with performing time-frequency localization of the signal to be analyzed (i.e,. the "input signal"). Time-frequency

localization refers to the analysis of a portion of the frequency spectrum of the input signal over a selected time window. As will become apparent from the description in this specification, time-frequency localization of an input signal enables data processing techniques to be applied to the signals for a wide range of purposes.

a. Conventional analog filtering

In the time domain, frequency-windowing is done by convolving a time-domain window filter function with the input signal; in the frequency domain, the frequency-windowing is done by multiplying the spectrum of the input signal with the frequency-domain transfer function of the filter function. Typical filter functions include low-pass filters (e.g., the Shannon sampling function) and band-pass filters. Through use of such filters, a bandwidth limited signal $f_\Omega(t)$ (i.e., zero amplitude for all frequencies above a limit Ω) may be decomposed into the sum of a low-frequency component with a series of non-zero frequency bands. An expression for such a decomposition is as follows:

$$f_\Omega(t)=f_{\Omega,\omega_0}(t)+g_{\Omega,1}(t)+\ldots+g_{\Omega,N}(t) \qquad [1]$$

where $f_{\Omega,\omega 0}$ is the low-pass filtered ($\omega<\omega 0$) component of the input signal $f_\Omega(t)$, where $g_{\Omega,i}(t)$ is the band-pass filtered signal for the ith frequency band, and where $\omega_N = \Omega$. An ideal low-pass filtered component $f_{\Omega,\omega 0}$ corresponds to the time-domain convolution of the well-known Shannon

sampling function with the input signal. Ideal band-pass filtering may be performed by time-domain convolution of a filter function of the type:

$$\frac{\sin\omega_n t - \sin\omega_{n-1} t}{\pi t} \qquad [2]$$

with the time-domain input signal, ω_n and ω_{n-1} being the upper and lower limits of the frequency band.

Each of the low-pass and band-pass filter functions provide ideal frequency localization of the input signal $f_\Omega(t)$, such that each expression $f_{\Omega,\omega 0}$ and $g_{\Omega,i}(t)$ provide precise information regarding the frequency spectrum of input signal $f_\Omega(t)$ within its respective frequency band. However, the time localization provided by this decomposition is quite poor, as these filters do not provide very precise information about when (in the time domain) a signal behaves differently within certain frequency ranges. As many important real-world signals include brief periods of time of rapid transient change, analysis of signals decomposed in this manner will not be able to account for the time at which such transient behavior occurs. For signal analysis where time localization is important, conventional Fourier analysis techniques therefore falls short of the need.

b. Time and frequency windowing

Certain conventional signal analysis techniques have addressed this problem by time-windowing the input signal,

thus allowing time-localization as well as frequency localization. According to these techniques, a window function h(t) is applied to the input signal f(t) to window the input signal near a specified point in time t=b. This windowing may be considered by the integral transform:

$$\int_{-\infty}^{\infty} f(t)\overline{h(t-b)}\,dt \tag{3}$$

where the bar over the function h(t-b) denotes complex conjugation. For a real-valued even windowing function h(t), this windowing process corresponds to convolution, allowing treatment of the windowing function h(t) as a filter. For example, if h(t) is the Shannon sampling function, the windowing process of the above equation will apply a low-pass filter to the input signal. However, it has been observed that the slow decay of the Shannon sampling function over time results in a very imprecise time-domain windowing operation.

Those windowing functions h(t), for which the square of the magnitude have finite first and second moments and finite energy, (i.e., that decay sufficiently rapidly at infinity) will produce a time-window having a "center" t_* and a "radius" Δ_h. The center t_* may be calculated as a mean value, while the radius may be calculated as the standard deviation of the windowing function around the mean t_*. For a windowing function of radius Δ_h, the width of the time-window will be $2\Delta_h$, commonly referred to as the RMS duration of the windowing function h(t). In the frequency domain, if the square of the magnitude of the

Fourier transform $\hat{h}(\omega)$ of the windowing function h(t) has finite first and second moments, the frequency domain windowing function $\hat{h}(\omega)$ will have a center ω_* and a width $2\Delta_{\hat{h}}$, calculated in a manner similar to the mean and standard deviation (doubled) of the frequency domain function $\hat{h}(\omega)$; the width $2\Delta_{\hat{h}}$ is usually called the RMS bandwidth of the windowing function h(t). If the RMS duration Δ_h is finite, then the time-domain windowing function h(t) is a time-window; similarly, if the RMS bandwidth $2\Delta_{\hat{h}}$ is finite, the frequency-domain windowing function $\hat{h}(\omega)$ is a frequency window.

Referring back to the ideal low-pass and band-pass filter time-domain functions noted above, it will be readily apparent that their first moment is infinite, meaning that neither of these ideal filter functions can serve as time windows if used as windowing functions h(t). However, the frequency domain representations of the ideal low-pass and band-pass filter functions have finite RMS bandwidth $2\Delta_{\hat{h}}$, so that these filters provide ideal frequency-localization as is evident from their ideal nature.

As discussed above, accurate analysis of real-world signals containing transient behavior requires both time-localization and frequency-localization. The Uncertainty Principle has identified those windowing functions h that provide both time windowing and frequency windowing as those functions that satisfy the following inequality:

$$\Delta_h \Delta_{\hat{h}} \geq \tfrac{1}{2} \qquad [5]$$

It has also been previously found that the only types of windowing functions h(t) that achieve the lower bound of the Uncertainty Principle are those of the form:

$$h(t) = ce^{jat}e^{(t-b)^2/4\alpha}$$

[6]

for some constants a, b, c and α with $\alpha > 0$ and $c \neq 0$.

Further indication of the presence of a time-frequency window by any windowing function h(t) may be obtained through the Parseval identity. The generalized windowing function h(t) noted above corresponds to:

$$\int_{-\infty}^{\infty} f(t)\overline{h(t-b)}\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\, e^{jb\omega}\overline{\hat{h}(\omega)}\,d\omega$$

[7]

With reference to the left side of equation [6], the time window is given by:

$$[b + t^* - \Delta_h,\ b + t^* + \Delta_h] \qquad [8]$$

such that the time window that is centered on $t = t_*$ is shifted by the parameter b; the time window also has a radius of Δ_h in the time dimension. Similarly, with reference to the right-hand side of equation [6], the frequency window is given by:

$$[\omega^* - \Delta_{\hat{h}},\ \omega^* + \Delta_{\hat{h}}] \qquad [9]$$

and is thus centered at $\omega = \omega_*$ with width $2\Delta_{\hat{h}}$. Figure 1a illustrates the location of time-frequency window 2 of equation [6] in a time-frequency coordinate system.

For causal real-valued window functions, where:

$$\hat{h}(-\omega) = \overline{\hat{h}(\omega)}$$

the function $|\hat{h}(\omega)|$ is an even function, so that the center ω_* is located at $\omega = 0$ and the frequency window of equation [9] becomes

$$[-\Delta_{\hat{h}}, \Delta_{\hat{h}}] \qquad [10]$$

However, while the time-frequency window of the filtering function h(t) may be moved along the time axis of Figure 1a by changing the value of b, the window is fixed in frequency at the center frequency ω_* or, in the case of real-valued even functions as noted above relative to equation [10], is fixed at a center frequency $\omega_* = 0$, as shown in Figure 1b. This fixation in frequency of the time-frequency window in limits the usefulness of the windowing process of equation [3].

c. The Short-Time Fourier Transform (STFT)

Rudimentary Fourier theory indicates that translation in the time-domain corresponds to a phase-shift in the frequency domain, and conversely that a phase-shift in the time-domain corresponds to a translation in the frequency domain. Accordingly, a phase shift in the windowing process of equation [3] should allow for sliding of the frequency window along the frequency-axis.

Considering a real-valued window function $\phi(t)$ which serves as a low-pass filter (i.e., $\hat{\phi}(\omega=0) = 1$), and for which $\phi(t)$, $|t|^{1/2}\phi(t)$ and $t\phi(t)$ are in L^2, the short-time Fourier transform of $\phi(t)$ is defined by:

$$(G_\phi f)(b,\xi) := \int_{-\infty}^{\infty} f(t)\, e^{-j\xi t}\phi(t-b)\, dt \qquad [11]$$

for $f \in L^2$. The short-time Fourier transform (STFT) of equation [11] is also referred to in the literature as the Gabor transform. Applying the Parseval identity results in the following expression for the STFT, from which the center and radii of the time and frequency windows are apparent:

$$(G_\phi f)(b,\xi) = \frac{e^{-j\xi b}}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\, e^{jb\omega}\overline{\hat{\phi}(\omega-\xi)}\, d\omega \qquad [12]$$

The short-time Fourier transform of equations [11] and [12] provide the improvement over the simple time-windowing process of equation [3] in that the frequency window function $\hat{\phi}(\omega)$ is able to slide along the frequency axis. The frequency window for a windowing function $\hat{\phi}(\omega)$ that otherwise (i.e., for $\xi=0$) has its center ω_* at $\omega=0$, is now localized to frequencies near $\omega = \xi$:

$$[\xi - \Delta_{\hat{\phi}},\ \xi + \Delta_{\hat{\phi}}] \qquad [13]$$

Similarly, where the center t_* of the windowing function $\phi(t)$ is also otherwise (b=0) located at the origin, the time window is now localized near t=b as follows:

$$[b - \Delta_\phi,\ b + \Delta_\phi] \qquad [14]$$

The STFT thus allows for sliding of the time-frequency localization windows both in time and in frequency, merely by changing the values of the phase-shift factors b, ξ, respectively. Figure 1c illustrates the position in time-frequency space of two time-frequency windows 5_0, 5_1, having phase-shift factors (b_0, ξ_0), (b_1, ξ_1), respectively. As a result, the STFT allows the lowpass window function to perform bandpass filtering by changing the value of ξ. However, as is evident in Figure 1c, the widths of the windows are invariant with changes in time-shift factor b or frequency-shift factor ξ. Accordingly, while analysis is improved through use of the short-time Fourier transform, inaccuracies due to undersampling have been observed for those transient periods of time in which rapid changes (i.e., amplitudes over a wide range of frequencies) exist. Accordingly, not only is the ability to slide the localization windows in both time and frequency desired, but it is also desirable to allow for scaling of the window widths as a function of time or frequency.

3. Wavelet analysis

a. Theory

Wavelet analysis techniques address the need for time-frequency localization windows of flexible size, primarily by introducing a scale parameter that is mapped to the frequency variable. As a result, the scale parameter will change the widths of the windows as a function of frequency, thus changing the aspect ratio of the windows with changes in frequency. Since the Uncertainty Principle requires the area of the windows to remain above a certain value, the time window width decreases and the frequency window width increases proportionally as the center frequency ξ increases. The narrowing of the time window and widening of the frequency window for high frequency environments more precisely detects and analyzes these high frequency portions of input signals.

The basic wavelet transform is known as the integral wavelet transform, or IWT. The IWT includes a scale parameter a in its definition as follows:

$$\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t)\,\overline{\psi\left(\frac{t-b}{a}\right)}\,dt \qquad [15]$$

As such, the window function ψ narrows with changing values of the scale parameter a, such that the time width of ψ decreases with decreasing a. The windowing function $\psi(t)$ used in the IWT of equation [15] is to be real-valued as before, but the IWT constraints also require $\psi(t)$ to be a

bandpass filter rather than a low pass filter, such that its Fourier transform $\hat{\psi}(\omega=0) = 0$, stopping at least zero frequency components of the signal. Since the windowing function $\psi(t)$ is real-valued, its Fourier transform $\hat{\psi}(\omega)$ satisfies:

$$\hat{\psi}(-\omega) = \overline{\hat{\psi}(\omega)}$$

[16]

so that $|\hat{\psi}(\omega)|$ is an even function. Because only nonnegative frequencies are of interest, and since $\psi(t)$ is a band-pass filter, the Fourier transform $\hat{\psi}(\omega)$ need only be considered as a frequency window in the frequency domain $[0, \infty)$, with the centers and widths of the frequency window function $\hat{\psi}(\omega)$ being modified as a result. For a windowing function $\psi(t)$ in L^2, for which $|t|^{1/2}\psi(t)$ and $t\psi(t)$ are also in L^2 such that $\psi(t)$ is real-valued, and where $\hat{\psi}(\omega=0) = 0$, the one-sided (i.e., nonnegative) frequency window center ω^*_+ as a function on the domain $[0, \infty)$ is defined as:

$$\omega^*_+ := \frac{\int_0^\infty \omega |\hat{\psi}(\omega)|^2 d\omega}{\int_0^\infty |\hat{\psi}(\omega)|^2 d\omega}$$

[17]

and the one-sided radius of $\hat{\psi}(\omega)$ is defined as:

$$\Delta_{\hat{\psi}}^{+} := \sqrt{\frac{\int_{0}^{\infty} (\omega - \omega_{+}^{*})^{2} |\hat{\psi}(\omega)|^{2} d\omega}{\int_{0}^{\infty} |\hat{\psi}(\omega)|^{2} d\omega}}$$

[18]

This allows the generation of an integral wavelet transform (IWT) using a normalization factor $a^{-1/2}$ based upon the scale parameter a, which scales the time-width of the window as a function of frequency. For a windowing function $\psi(t)$ that satisfies the conditions for equations [17] and [18] above, the IWT is defined as follows:

$$(W_{\psi}f)(b,a) := \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt$$

[19]

For the IWT of equation [19], the bandpass window-function $\psi(t)$ is commonly referred to as the analyzing wavelet.

As is known in the wavelet analysis field, and given the foregoing discussion, it is important that the integral wavelet transform W_{ψ} allows for frequency localization where the width of the time window is mapped to the frequency domain, and in which the frequency window can slide along the frequency axis. For finite-energy real-

valued input signals f(t), and since ψ(t) is real, the following relationship holds:

$$\hat{f}(-\omega)\, e^{jb\omega}\overline{\hat{\psi}(-a\omega)} = \overline{\hat{f}(\omega)\, e^{-jb\omega}\hat{\psi}(a\omega)}$$

[20]

Through the Parseval identity, one can then derive the IWT as follows:

$$(W_{\psi}f)(b,a) = \frac{\sqrt{a}}{\pi} Re\int_{0}^{\infty} \hat{f}(\omega)\, e^{-jb\omega}\eta\left(a\left(\omega - \frac{\omega_{+}^{*}}{a}\right)\right) d\omega$$

[21]

for all f $\in L^2_R$, where ω_+^* is the one-sided center of $\hat{\psi}(\omega)$ on the domain [0, ∞), and where η is defined as follows:

$$\eta(\omega) := \overline{\hat{\psi}(\omega + \omega_{+}^{*})}$$

[22]

As noted above, it is desirable to map the scale parameter a to the frequency at which the time-frequency window is to be localized. Accordingly, the scale parameter a is mapped to the shift frequency parameter ξ as follows:

$$\xi := \frac{c}{a}$$

[23]

for some c > 0, where c is a calibration constant. Substituting for the scale parameter a defines the IWT as follows:

$$(W_\psi^* f)(b, \xi) := (W_\psi f)(b, a) = (W_\psi f)\left(b, \frac{c}{\xi}\right)$$

[24]

It is convenient to set $c = \omega_+^*$, so that:

$$(W_\psi^* f)(b, \xi) = \frac{\sqrt{a}}{\pi} \mathrm{Re} \int_0^\infty \hat{f}(\omega)\, e^{-jb\omega} \eta(a(\omega - \xi))\, d\omega$$

[25]

This produces a frequency window $\eta(a(\omega - \xi))$ that slides along the frequency axis with the value of ξ, having a range:

$$\left[\xi - \frac{1}{a}\Delta_\phi^+, \xi + \frac{1}{a}\Delta_\phi^+\right]$$

[26]

The width of this window thus increases at higher frequencies ξ, as reflected in smaller values of the scale parameter a. In terms of the scale parameter a, the width of the frequency-window is as follows:

$$\left[\frac{1}{a}(\omega_{+}^{*}-\Delta_{\phi}^{+}),\ \frac{1}{a}(\omega_{+}^{*}+\Delta_{\phi}^{*})\right]$$

[27]

where the frequency shifting term $\xi = \omega_{+}^{*}/a$, such that the frequency-width of the frequency window increases with increasing frequency ξ (decreasing values of a). Along the time-axis, the time window of the IWT of equation [19] is given by:

$$[b+at^{*}-a\Delta_{\psi},\ b+at^{*}+a\Delta_{\psi}]$$

[28]

As a result, the width of this time-window is $2a\Delta_{\psi}$, which decreases at higher frequencies ξ (and lower values of a), and which increases at lower frequencies ξ. For the transform of equation [15], Figure 1d illustrates three time-frequency windows 6_0, 6_1, 6_2, with varying translation factor pairs (b_0, ξ_0), (b_1, ξ_1), (b_2, ξ_2), respectively. As is evident in Figure 1d, both the time-width and frequency-width of windows 6 vary with varying shift frequency ξ, such that the time-width decreases and the frequency-width increases with increasing ξ.

Accordingly, it should now be apparent to those in the art that wavelet analysis is capable of providing increased accuracy analysis of signals, particularly those including transient components, considering that, for higher frequency analysis, the width of the time window decreases and the width of the frequency window increases. This ensures adequate sampling of the input signal function, and also allows for determination of the exact time in the signal at which the transient event occurred.

As is known in the art of wavelet analysis, however, the definition of the proper wavelet function $\psi(t)$ is of great importance. Various specified analyzing functions have been used in wavelet analysis, with the selection of the function made according to computability, or according to attributes of the signal under analysis.

b. Prior wavelets

Wavelet signal analysis has been applied to signals produced in seismic exploration for oil and gas, as described in U.S. Patent No. 4,599,567. This reference describes a wavelet analysis technique using Morlet's wavelet as the analyzing wavelet. Morlet's wavelet is a sinusoid limited by a Gaussian probability envelope to a finite duration; the envelope may or may not be skewed toward either the leading or lagging edge of the time-domain envelope of the wavelet, as desired for the type of signal under analysis. This reference also discloses circuitry for performing the wavelet analysis of an incoming signal using such analyzing wavelet.

Another finite duration analyzing wavelet was proposed by Yves Meyer. This analyzing wavelet is substantially a

finite duration uniform magnitude level over the wavelet window, analogous to the Shannon sampling function.

Other wavelets have been proposed which are not expressible by a mathematical formula, but instead are utilized as numeric wavelets. These wavelets include the Battle-Lemanné wavelet, which is a spline function of infinite duration, and the Daubechies wavelet, which is a fractal wavelet of finite duration. The lack of explicit formulae for these wavelets limit their applicability for rapid and accurate computer implementation of the wavelet analysis technique in computer hardware.

Another previously published wavelet is the Chui-Wang wavelet, which is a wavelet of finite duration but which may be expressed in an explicit formula.

The support for each of the prior wavelets noted above is over an unbounded interval. However, since real-world problems require the application of the wavelets to bounded intervals, wavelet analysis of input signals using these prior wavelets result in errors of the type commonly referred to as "boundary effects".

Figure 2 graphically illustrates the reason for boundary effects arising from such conventional wavelets. Conventional wavelet 7 illustrated in Figure 2a is first-order spline-wavelet that is based on a function ψ having moments with the following properties (for $i = 0, 1, \ldots m-1$, and some $m \geq 1$):

$$\int_{-\infty}^{\infty} t^{i} \psi\left(\frac{t-b}{a}\right) dt = 0$$

[29]

where a, b are scaling parameters as discussed above. The conventional wavelet 7 of Figure 2a is not orthogonal over a bounded interval [c, d], however, meaning that:

$$\int_c^d t^i \psi\left(\frac{t-b}{a}\right) dt \neq 0$$

[30]

for all integers i > 0.

Figure 2 graphically illustrates the performance of the IWT using wavelet 7 having the above noted properties at a point in the time series which happens to be at or near the boundary of an interval [c, d]. Data points f_2, f_3, f_4 correspond to input signal samples within the interval, which are plotted against wavelet 7 in Figure 2; in this example, for purposes of explanation, the input signal sample data closely matches the shape of wavelet 7 within the interval [c, d]. The position of wavelet 7 corresponds to the position, in performing the IWT, of the sample point of interest at the boundary value t=a. Since wavelet 7 at this position requires support outside of the interval [c, d] for which input signal data exists, the zero values that must be assumed for the data points f_0, f_1 outside of interval [c, d] necessarily fail to match wavelet 7. This will result in an non-zero result for the IWT, even where the input data signal within the interval [c, d] exactly matches wavelet 7. As is well known in the field of signal processing, this inaccuracy due to wavelet 7 requiring support outside of the bounded interval [c, d] is made manifest by boundary effects at the edges of the bounded interval, since the unbounded wavelet 7 fails to accurately represent the series of actual input signal sample data.

In the field of video image compression and analysis, boundary effects greatly affects the quality of the image displayed after compression and decompression. This is because the boundary effects will appear as false image data at the edges of pixel blocks corresponding to bounded intervals, and also at the edges of the image (even if no subdivision of the image is performed). The inaccuracies due to boundary effects also limit the ability to magnify an image when displayed, considering that the magnification will make the boundary effect errors to become even more visible.

c. The boundary-spline-wavelet

By way of further background, a bounded interval wavelet is described in Chui and Quak, "Wavelets on a Bounded Interval", Numerical Methods of Approximation Theory, Volume 9 (Dec. 1992), pp. 53-75, incorporated herein by this reference. This wavelet, which has an explicit formula, is not only a function of the time variable t, but is also a function of the position of the sample within the interval [c, d], so as to account for boundary effects. In effect, sample locations near the boundaries of the interval will correspond to different wavelet shapes than will sample locations within the interval that are away from the boundaries. Boundary effects are eliminated as the boundary wavelets do not require support outside of the interval.

Referring now to Figures 3a through 3d, an exemplary set of four first-order wavelets 8 according to the Chui-Quak boundary-spline-wavelet approach are illustrated. Figure 3a illustrates the shape of "boundary" wavelet 8_a for a sample location near the boundary t=a of the interval

[a, b], while Figure 3d illustrates the shape of boundary wavelet 8_b for a sample location near the boundary t=b of the interval [a, b]. Figures 3b and 3c each illustrate the shape of "inner" wavelets 8_i for sample locations within the interval [a, b] away from the boundaries. As is evident from Figures 3a through 3d, boundary wavelets 8_a, 8_b, have different shapes than inner wavelets 8_i (which have the same shape as one other). As is further evident from Figures 3a through 3d, neither inner wavelets 8_i nor boundary wavelets 8_a, 8_b require support outside of the interval [a, b], or:

$$\int_a^b t^i \psi\left(\frac{t-b}{a}\right) dt = 0 \qquad [31]$$

for i = 0, 1, . . ., m-1, and for some m ≥ 0, for the entire set of wavelets 8 of Figures 3a through 3d. Accordingly, application of the set of boundary-spline-wavelets 8 to actual real-world data, for which the time interval is necessarily bounded, will not produce boundary effect artifacts.

Other boundary-wavelets are known, as described in Daubechies, "Two recent results on wavelets : Wavelet bases for the interval and biorthogonal wavelet diagonalizing the derivative operator", Recent Advances in Wavelet Analysis, Schumaker and Webb, ed. (Academic Press, 1993), pp.237-58. These wavelets are not spline functions, and do not have explicit formulae. As a result, it is believed that these wavelets are of limited effectiveness in video image compression and decompression.

4. Objects of the invention

It is therefore an object of the present invention to apply wavelet analysis tools to the task of video image compression for storage and transmission.

It is a further object of the present invention to provide an apparatus for performing video image compression according to a selected wavelet.

It is a further object of the present invention to provide an apparatus for receiving a compressed video image and to decompress the same for real-time playback of the stored or communicated video image information.

It is a further object of the present invention to provide such a method of compression and decompression such that the compressed image may be played locally at real-time.

It is a further object of the present invention to provide such a system and method which allows for high accuracy magnification of the decompressed image, with much reduced incidence of edge effects.

It is a further object of the present invention to provide such a system and method which provides a high degree of compression.

It is a further object of the present invention to provide such a system and method which can utilize dynamic compression on a frame-by-frame basis, such that high frequency frames may be compressed to different ratios than low frequency frames.

It is a further object of the present invention to provide such a system and method which facilitates interactive display of a motion image, including insertion, editing and repetitive display.

It is a further object of the present invention to provide such a system and method which provides the ability for slow display systems to skip certain frames so that a motion picture can be displayed on a real-time basis by such slower systems, although with fewer frames per second.

It is a further object of the present invention to provide such a method and system which allows for division of an image into several portions for purposes of compression and communication or storage, with subsequent display of the full image after decompression.

It is a further object of the present invention to provide such a method and system which allows for higher lossy compression ratios by further quantization, as useful in compressing and decompressing high-quality 24-bit still images.

Other objects and advantages will be apparent to those of ordinary skill in the art having reference to the following specification, together with its drawings.

<u>Summary of the Invention</u>

The present invention may be implemented into a method and apparatus for compressing and decompressing video images. The compression/decompression system receives video image data in a suitable format, such as eight-bit Portable Grey Level format, and includes circuitry for decomposing each component of the image into low and high frequency components by way of boundary-spline-wavelet decomposition. The system also includes circuitry for quantizing the high frequency portions of the decomposition. The decomposition may be repeated until the desired compression ratio is obtained. Lossless compression may then be performed on the compressed image, following which the compressed data may be stored for archival purposes, or may be transmitted to a receiving station.

According to the present invention, in decompression mode, the system performs lossless decompression of the stored or communicated image data, according to the compression scheme utilized. Reconstruction of the image is then performed by way of the boundary-spline-wavelet approach, in a reverse fashion relative to the image compression. The result of the decompression is then formatted as desired for display.

Brief Description of the Drawings

Figures 1a through 1d are time-frequency plots illustrating the concept of time-frequency windowing, as described in the background of the invention.

Figure 2 is a graphical illustration of a wavelet with non-zero moments of any order relative to the interval [c, d], as applied to an input signal, and which is the source of boundary effects.

Figures 3a through 3d are graphical illustrations of an exemplary set of boundary-spline-wavelets as used in the preferred embodiment of the invention.

Figure 4a is an electrical diagram, in block form, of a system for producing and communicating or storing compressed video image information according to the preferred embodiment of the invention.

Figure 4b is an electrical diagram, in block form, of the image format converter in the system of Figure 4a.

Figure 5 is an electrical diagram, in block form, of a compressor according to the preferred embodiment of the invention.

Figure 6 is a flow diagram illustrating a method of compressing video image data according to the preferred embodiment of the invention.

Figure 7 is a detailed flow diagram illustrating a method of performing the boundary-spline-wavelet

decompression according to the preferred embodiment of the invention.

Figure 8 is a series of frequency domain plots illustrating the decomposition of the video image according to the preferred embodiment of the invention.

Figures 9a through 9c are matrices illustrating the effect of decomposition of the video image according to the preferred embodiment of the invention.

Figure 10 is a flow diagram of a preferred matrix algebra technique for performing the boundary-spline-wavelet decomposition of Figure 7.

Figure 11 is a chart illustrating the relationship of the matrices in the process of Figure 10.

Figures 12a through 12e illustrate an example of a video image decomposed according to the preferred embodiment of the invention.

Figures 13a through 13c illustrate the quantization of the decomposed video image of Figures 12a through 12e according to the preferred embodiment of the invention.

Figure 14 is a diagram of the construction of a compressed video image frame for transmission according to the preferred embodiment of the invention.

Figure 15a is an electrical diagram, in block form, of a decompressor according to the preferred embodiment of the invention.

Figure 15b is an electrical diagram, in block form, of a format converter useful in the decompressor of Figure 15a.

Figure 16 is a flow chart of a process for decompressing video image data according to the preferred embodiment of the invention.

Figure 17 is a detailed flow chart of a process for reconstructing video image data that were decomposed according to the boundary-spline-wavelet approach of the preferred embodiment of the invention.

Figure 18 is a chart illustrating the relationship of the matrices as used in a preferred matrix algebra routine for the reconstruction of Figure 17.

Figure 19 is an electrical diagram, in block form, of a decompressor system according to an alternative embodiment of the invention.

Detailed Description of the Preferred Embodiment

1. Video image data communication system

Referring now to Figure 4a, the construction of a video communication system 8 according to the preferred embodiment of the invention will now be described in detail. In this embodiment, system 8 is specifically designed for communicating compressed motion picture data; it is of course contemplated that system 8 may alternatively be used to communicate compressed data representative of a series of still video images. In addition, as will be described in further detail hereinbelow, system 8 may alternatively or also be used for archival storage of motion picture or still video image data, and the subsequent retrieval and display thereof.

As shown in Figure 4a, in the context of video data communication, the transmission end of system 8 includes video source 12 and compression system 10, while the receiving end of system 8 includes decompression system 20 and video display 26. As illustrated in Figure 4a, video source 12 may include digital video source 12d which may be a conventional device such as a CD-ROM drive, scanner, digital electronic network connection, or similar unit, or alternatively may be a computer storage unit such as a disk that contains digital video information. Video source 12 may also or instead include analog video source 12a, which may be a video camera, VCR unit, television broadcast or cable receiver, or another conventional source of analog video information. In any case, video source 12 provides digital or analog signals indicative of the images to be communicated or stored by the system of Figure 4a.

Communication network system 15 is a conventional analog transmission or electronic digital communications network, or both an analog and digital network when including the appropriate clusters, analog-to-digital and digital-to-analog converters, and other necessary apparatus. Network 15 may be realized according to any conventional technology, including hard-wired cable, fiber optic cable, broadcast or satellite transmission, and the like. It is further contemplated that network 15 may be implemented merely by the physical transportation of portable media 22' such as floppy diskettes, CD-ROMs and the like. Regardless of the implementation, network 15 is connected between compression system 10 and the input of decompression system 20 to communicate compressed video image data therebetween.

It is specifically contemplated that the video data transmission effected from compression system 10 may be of the broadcast type, such that a single transmission end (source 12 plus compression system 10) may communicate simultaneously or in sequence to multiple receiving ends (decompression system 20 and display 26). For example, compression system 10 may be located at a television or movie studio, or at a local cable television system "head-end", with multiple receiving ends (decompression system 20 and display 26) located at homes, offices, or local theaters, depending upon the particular transmission being effected.

Figure 4a also illustrates the optional use of system 8 in the context of the archival storage of motion picture or still video image data, and its later retrieval and display thereof. Disk storage unit 22 is coupled to receive the compressed video data from compression system 10 for storage. An example of disk storage unit 22 is a

large disk (e.g., having capacity on the order of 1000 Gigabytes); alternatively, or in addition to the large disk unit, disk storage unit 22 may be associated with a mainframe or supercomputer to provide services such as on-line collection of video images into a library form. Disk storage unit is also coupled to decompression system 20, which is operable in this context to receive compressed video data therefrom for display on video display 26. The subsequent retrieval of the archived information in disk 8 may be made via a digital communications network (such as network 15 described hereinabove), or alternatively by way of a portable data storage medium such as a tape or diskette.

Compression system 10 includes data format converter 14 that is functionally connected between video source 12 and compressor/coder 16. Format converter 14 is of conventional construction as used to convert the format of the video image data from video source 12 into the format suitable for compression by compressor 16. In the example described hereinbelow, the format utilized by compressor 16 is eight-bit Portable Grey Level (PGM), although fewer than eight bits may be used in the PGM format, depending upon the available hardware devices and architecture. For color images, the format is RGB-PGM, where each of the red, green and blue color components is separately stored and communicated in PGM format. The PGM image format will be described in further detail hereinbelow. Accordingly, format converter 14 converts the image data from video formats such as PCX, IMG, GIF, TIF, RLE, NTSC, PAL, and the like into PGM or RGB-PGM format. Of course, if the output from video source 12 is already in PGM or RGB-PGM format, format converter 14 is unnecessary, and compressor 16 may directly receive the data from video source 12.

Referring now to Figure 4b, the construction of format converter 14 according to the preferred embodiment of the invention will be described in detail, for the example where the output provided by format converter 14 is in RGB-PGM format. As noted above, format converter 14 can receive video data from either analog or digital video sources 12a, 12d, respectively, and as such two input ports are provided to format converter 14 along with two functional devices to reformat the received data into the desired format.

In the example of Figure 4b, digital video image data presented by digital video source 12d is received by format decoder and color expansion device 21. The format decoder portion of device 21 decodes the data from the format presented by source 12d, such format being PCX, IMG, TIF, GIF, RLE, YUV, etc., into a color matrix, or color palette table, representation of the image signal. Format decoder circuitry is well known in the art, such as described in Rimmer, Supercharged Bitmapped Graphics (Windcrest/McGraw Hill). This decoding is followed by a color expansion operation performed by device 21, such that the output therefrom is in raw RGB format, with each pixel represented by successive bytes of red, green and blue color components. Conventional circuitry for performing such color expansion is readily available in the field, including such devices as color palette RAMs. The output of format decoder and color expansion device 21 is presented to color separator 25.

On the analog side, analog video information is presented by analog video source 12a to analog screen grabber and digitizer device 23. Device 23 first converts the analog input data from its NTSC or PAL format, by the screen grabber capturing the input screen data. Device 23

then digitizes each pixel of the captured screen, in the conventional manner. Conventional circuitry for performing the screen grabbing and digitizing functions may be used to implement device 23. The output of device 23 is also in the raw digital, or binary, RGB format as presented by device 21.

Color separator 25 is a conventional digital filter for separating the raw binary RGB signal presented by either of devices 21, 23 into RGB-PGM format. As is well known in the art, the RGB-PGM video data format includes, for each pixel, three eight-bit components. These three eight-bit components correspond to the intensity of the red, blue and green colors, respectively, to be displayed for that pixel. Of course, other byte widths may be used to digitally represent the intensity of each color component.

The RGB-PGM format used in this preferred embodiment of the invention decomposes the input video image into three screen representations, one for each of the red, blue and green color components. Each pixel location in the input image is thus represented by a pixel value in each of the red, blue and green decompositions, indicating the intensity of that color component at that pixel location of the image. In the eight-bit PGM format, the intensity of each color component for each pixel can thus range from 0 (black, or no intensity) to 255 (full intensity). The eight-bit PGM format is particularly convenient for processing by microprocessors or computers using ASCII coding and programming environments, as the fundamental storage unit is the byte in these environments.

The three channels of RGB-PGM data produced by color separator 25 is then presented to compressor 16, either

sequentially or in parallel, for compression as will be described hereinbelow. According to this embodiment of the invention, each of the R, G, B components are compressed, decompressed, and otherwise processed separately from the other components for that image.

Of course, if the data presented by video source 12, specifically digital video source 12d, is already in RGB-PGM format, format converter 14 is not necessary in the system of Figure 4a.

Compressor 16 includes the necessary compression circuitry, an example of which is described in detail hereinbelow, for compressing the formatted digital data according to the boundary-spline-wavelet technique that is also described in detail hereinbelow. Compressor 16 may also include coding circuitry for formatting the compressed data into the suitable format for communication over network 15 (or storage in disk 22). As will also be described in further detail hereinbelow, specific information regarding attributes of the communicated or stored information may be specifically included in the data coded by compressor 16.

Similarly, decompression system 20 includes decompressor 18 coupled to network 15 (or to disk 22, as the case may be). Decompressor 18 receives the transmitted or stored image data, reformats it into a form suitable for decompression, if necessary, and decompresses the data. Decompressor 18 in this embodiment of the invention communicates the decompressed image data to format converter 24, which converts the decompressed data to the appropriate format for display by display 26.

As illustrated in Figure 4a, display 26 may be implemented as a digital display 26d, to which digital data may be directly applied thereto; alternatively, display 26 may be implemented as a conventional analog display, with the appropriate NTSC or other analog video format data applied thereto. According to this example, the output data from decompressor 18 is in PGM or RGB-PGM format, and thus format converter 24 will convert the PGM data into PCX, IMG, GIF, TIF, RLE, NTSC, PAL, RGB, or another display format. Of course, if digital display 26d is used and is capable of receiving and directly displaying PGM format data, format converter 24 is unnecessary.

2. Construction of the compressor

Referring now to Figure 5, the construction of compressor 16 according to the preferred embodiment of the invention will now be described in detail. It is, of course, contemplated that other architectural arrangements of circuitry may be used in the compression of video data according to the present invention. Specifically, it is contemplated that a conventional general purpose computer system may be capable of performing the compression of the video image data according to the present invention. However, the example of Figure 5 incorporates a preferred embodiment of the circuitry for performing the data compression functions to be described herein.

In this example, compressor 16 preferably includes one main RGB-PGM channel controller 28 and three substantially identical channel compressor subsystems 29_R, 29_G, 29_B. Three channel compressor subsystems 29_R, 29_G, 29_B are provided according to this embodiment of the invention, considering the separation of the input video data into the

three component channels by format converter 14 described hereinabove. Compressor 16 also preferably includes data flow interface 39, for receiving data from each of channel compressor subsystems 29 and presenting the same to network 15.

Main controller 28 receives the three channels of RGB-PGM data from format converter 14, and forwards each channel of data separately to the appropriate channel compressor subsystem 29_R, 29_G, 29_B. Alternatively, compressor 16 may be implemented to have only a single channel compressor subsystems 29 which processes each channel of data sequentially, in which case main controller 28 would control the sequential forwarding of image data to the single compressor subsystem 29.

Main controller 28 is connected by way of the appropriate bidirectional buses and control lines to control the functions within compressor 16. In addition to its control of subsystems 29, main controller 28 also controls the timing, feedback and sending operation of compressor 16, including control of the data flow interface 39. As such, main controller 28 is preferably a general purpose programmable microprocessor or other central processing unit (CPU) of sufficient computational power and capacity to process some or all of the image data and to control the performing of the image compression functions to be described hereinbelow. It is contemplated that microprocessors having performance levels similar to or greater than those of the 80486 type (including those available from Intel Corporation or Cyrix Corporation), of the 68040 type (including those available from Motorola), and of the SPARC processor type (available from Texas Instruments Incorporated or Sun Microsystems, Inc.) will be suitable for use as main controller 28 in compressor 16.

The construction of channel compressor subsystems 29 will now be described in detail relative to the construction of subsystem 29_R as shown in Figure 5; it is contemplated that the other subsystems 29_R, 29_G will be similarly constructed. Each channel compressor subsystem 29 according to this embodiment of the invention is specifically designed to perform the functions of boundary spline wavelet decomposition, quantization and lossless compression, as used in the compression operation according to this embodiment of the invention.

Channel compressor subsystem 29 according to this embodiment of the invention includes the circuit functions of digital matrix process 30, timing circuit 37, quantization processor 32, lossless compressor 34, program data memory 35, and multiple memory banks 36 for storage of image data.

Timing circuit 37 performs the functions of receiving the PGM format channel data from main controller 28 and forwarding the received data to memory banks 36 in its subsystem 29. In addition, timing circuit 37 controls the timing and feedback among the other functional components of channel compressor subsystem 29, including the matrix operations performed by digital matrix processor 30, quantization performed by quantization processor 32, lossless compression performed by lossless compressor 34, and accesses to memory 35, 36.

Digital matrix processor 30 is a processing circuit of conventional architecture that is specifically suitable for performing vector and matrix operations as used in the decomposition of image data according to the preferred embodiment of the invention, as will be described in detail hereinbelow. As will be apparent from the description

below, these operations include the retrieval of pre-calculated matrices from program data memory 35, the matrix operations utilized in performing boundary-spline-wavelet decomposition of the video channel data, and storage of the results in memory banks 35. Examples of presently available components suitable for use as digital matrix processor 30 include digital signal processors such as the i860 processor available from Intel Corporation and the TMSC40 digital signal processor available from Texas Instruments Incorporated, and also general purpose microprocessors such as those of the 80386 and 80486 families available from Intel Corporation, and of the 68030 and 68040 families available from Motorola.

Quantization processor 32 is a logic circuit for filtering the data corresponding to decomposed images in order to achieve the desired compression ratio. It is contemplated that conventional processing circuitry or custom logic circuitry for performing this function in the manner described hereinbelow, will be readily apparent to one of ordinary skill in the art. According to this preferred embodiment of the invention, quantization processor 32 may be selectably controlled to perform the quantization process according to various selectable modes. These modes are selectable by way of quantization mode register 31, which stores a three digit code defining the type of quantization to be performed by quantization processor 32. An example of the codes storable in quantization mode register 31 and their corresponding quantization modes are as follows:

0	:	No quantization
1	:	thresholding
2	:	scalar quantization
3	:	JPEG quantization (i.e., using tables)
4-7	:	reserved for other quantization modes (e.g., vector quantization)

Lossless compressor 34 may be implemented by way of a conventional digital signal processor such as the TMSC40, programmed in such a manner as to perform lossless decompression upon the results of the quantized output from quantization processor 32. The lossless decompression performed by lossless decompressor 34 is according to the desired conventional technique, such as Huffman encoding, adaptive Huffman encoding, arithmetic encoding, LSQ coding, and the like. Alternatively, lossless compressor 34 may be implemented as a custom logic circuit for providing this function. The output of lossless compressor 34 is preferably compressed data for the channel (R, G, B) in bitstream format, for application to data flow interface 39.

Data flow interface 39 provides an interface between compressor 16 and network 15, and as such must gather the bitstream output from lossless compressors 34 in each of the subsystems 29 and arrange the same into a suitable format for transmission. Data flow interface 39 also provides a feedback signal to main controller 28 upon transmission of a frame of compressed data, based upon which main controller 28 may commence the processing of the next image frame to be compressed.

The preferred format in which the output compressed data from data flow interface 39 is communicated over network 15 will be described in further detail hereinbelow.

This example of compressor 16 is intended to support the compression of high definition real-time true color video image data, where "true color" indicates the use of twenty-four bits of color information for each pixel, resulting in 16.7 million possible colors. The frame rate for compressor 16 is intended to be on the order of thirty frames per second, so as to support "real-time" video image compression.

As noted above, if the color and frame rate requirements are reduced from real-time true color video, it may be possible to implement compressor 16 as a single channel, i.e. with a single channel compression subsystem 29. In this implementation, color data could be compressed sequentially for the R, G and B components of the RGB-PGM input data, under the control of main controller 28.

3. Boundary-spline-wavelet video image data compression

Referring now to Figure 6, a method of compressing video image data according to the preferred embodiment of the invention will now be described in detail. It is contemplated that the architecture of compressor 16 of Figure 5 and described hereinabove is particularly suitable for the performing of this method, although it is further contemplated that other computer architectures and arrangements may alternatively be used to perform the process of Figure 6.

The flow chart of Figure 6 corresponds to the compression of a single frame of video image data. Accordingly, for the compression of a motion picture, the process of Figure 6 is performed sequentially for each frame of the motion picture. In the case of still image

compression, of course, the process of Figure 6 is performed for each image.

The process of Figure 6 begins with the conversion of the video image data for the frame into the desired format for compression which, according to this preferred embodiment, is the well-known PGM format. As discussed above relative to Figure 4a, this conversion is preferably performed by format converter 14, allowing compressor 16 to be dedicated to the compression process. As noted above, the PGM (Portable Grey Level) format expresses each picture element ("pixel") as a numerical value corresponding to its brightness; for eight-bit PGM, the values range from 0 to 255. Color images may be expressed in a mode referred to in the art as RGB-PGM, where a PGM image is provided for each of the red, green and blue color components of the image. It is believed that the RGB-PGM format is the most adaptable format for processing of color image data by way of the present invention, as it directly provides "true" color display information for high performance display systems, and is also readily adaptable for conversion to color look-up table ("color palette") display systems.

After conversion by format converter 14, the RGB-PGM image data is separated into separate R, G, B channels by main controller 28. This allows each of channel compression subsystems 29 to compress the image data in the manner described hereinbelow in parallel with one another. Of course, if a monochrome image is being compressed, only a single subsystem 29 is necessary; alternatively, if the compression rate allows, a color image may be compressed by sequentially compressing the separate R, G, B channel image data.

a. Boundary-spline-wavelet decomposition

After channel separation by main controller 28, a frame of PGM image data is stored in memory banks 36 in subsystem 29 in a row-based order, arranged from top left to bottom right of the image. Each PGM image frame is then processed by boundary-spline-wavelet decomposition, indicated by process 40 of Figure 6. Figure 7 is a detailed flow chart of process 40 according to the preferred embodiment of the invention, to which attention is now directed. The compression method of Figure 7 will be described relative to a single channel (R, G, or B), as it will be understood by those of ordinary skill in the art that a full color image will be compressed by the parallel or sequential compression of the other color channels in similar fashion.

The boundary-spline-wavelet decomposition of process 40 may operates upon individual blocks of pixels of the PGM frame, such as blocks of pixels that may be as small as eight-by-eight pixels, or as large as on the order of 1024-by-1024 pixels or greater. Process 48 indicates that the operation of each subsystem 29 will be performed upon the PGM frame in subdivided blocks. One of the important benefits of the present invention is that the size of the blocks defined in process 48 depends primarily upon the architecture of subsystems 29 in compressor 16 and of digital matrix processor 30 therein, but is not dictated by considerations of picture quality. This is because the boundary-spline wavelets used in the decomposition eliminate inaccuracies in the displayed data resulting from boundary effects.

Alternatively, if the computing capacity of digital matrix processor 30 and subsystems 29 is adequate, an

entire frame may be decomposed in process 40 without dividing the image into smaller blocks of pixels. The present invention will still provide important benefits due to the elimination of boundary effects and inaccuracies around the edge of the image that would otherwise be present if unbounded wavelets were used. In addition, regardless of whether the image is divided into smaller blocks, the elimination of boundary effects according to the present invention also enables magnification of the image to be performed after decompression, without the limitations and artifacts that would otherwise be present.

Following division of the frame into the appropriate image blocks, the decomposition of process 40 continues with processes 50, 52 in which boundary-spline-wavelet decomposition is performed upon the image block, according to this embodiment of the invention. As will be described in detail herein, according to the preferred embodiment of the invention, the decomposition of process 40 is performed utilizing the wavelets described in the Chui and Quak paper cited hereinabove and incorporated by reference hereinto, and illustrated in Figures 3a through 3d discussed above. Process 50 first decomposes each horizontal row of the image block into two equal numbers of components of low-frequency (low-pass) and high-frequency (band-pass), and is followed by process 52 in which each vertical column of each of the low-frequency and high-frequency results of process 50 are again decomposed into two equal numbers of components of low-frequency and high-frequency.

Referring now to Figure 8 in combination with Figures 9a through 9c, decomposition processes 50, 52 will now be described in further detail. Figure 8 illustrates a frequency domain representation of function $\hat{f}_N$, and which corresponds to image block 51 of Figure 9a that represents

the discrete function $f_N(x,y)$. The discrete function $f_N(x,y)$ is a functional representation of image block 51 in PGM format, such that each value "x" in Figure 9a is a digital value corresponding to image brightness at a position (x,y). In this example, image block 51 is eight pixels square.

As shown in Figure 8, process 50 of Figure 7 decomposes the function $\hat{f}_N$ into a low-frequency or low-pass component $\hat{f}_N$-1 and a high-frequency or band-pass component $\hat{g}_N$-1. According to the preferred embodiment of the invention, process 50 performs such decomposition for each horizontal row of image block 51, so that the result of process 50 is a matrix 53 that includes, for each row, a low frequency portion $f_{N-1}(x,y)$ and a high frequency portion $g_{N-1}(x,y)$. In other words, the decomposition of process 50 is performed considering the image data for each row of image block 51 as a one-dimensional spatial function in the x-dimension.

As discussed hereinabove, the decomposition of process 50 is performed according to boundary-spline-wavelet decomposition techniques, so that boundary effects are eliminated in the resulting frame. Accordingly, for each horizontal row of image block 51, the low frequency component $\hat{f}_N$ (or $f_{N-1}(x,y)$ in the spatial domain) in the corresponding row of decomposed image block 53 is a spline interpolation of the spatial data of image block 51. The high frequency component $\hat{g}_N$-1 (or $g_{N-1}(x,y)$ in the spatial domain) in the corresponding row of decomposed image block 53 corresponds to the boundary wavelets applied to the spatial data of that row of image block 51.

Referring back to Figure 5, according to the preferred embodiment of the invention, it is contemplated that the

decomposition of process 50 is performed by digital matrix processor 30 by way of a matrix algebra technique. According to this matrix technique, the decomposition of process 50 is intended to calculate the matrices c^{N-1}, d^{N-1} according to the following relationship:

$$\hat{f}_N=\hat{f}_{N-1}c^{N-1}+\hat{g}_{N-1}d^{N-1} \qquad [32]$$

where c^{N-1} and d^{N-1} are matrices of coefficients of the low-frequency and high-frequency components of the input signal, respectively.

Attention is directed to Figures 10 and 11 in combination for a detailed description of this technique. Figure 10 is a flow chart illustrating the matrix algebra procedure for performing processes 50 and 52 of Figure 7 (the algebra being the same in each case). Figure 11 is a chart illustrating the relationship among the decomposition matrices used in performing processes 50, 52 according to the flow chart of Figure 10.

As shown in Figure 10, the first step of the decomposition begins with the process 60 which interpolates the function f, for each horizontal row, using a B-spline function $\phi(x)$, to provide a matrix representation $\{c\}^N$ of the input image block 51 using the following relationship:

$$y_k=f(x_k)=\sum_{l=-m+1}^{2^N-1} c_l^N\phi_l^N(x_k)\,;k=1,2,\cdots,2^N-m+1 \qquad [33]$$

where m is the degree of the spline, where k is the number of sample points within the interval (i.e., the length of the row of image block 51, in pixels), and where the B-spline function $\phi_k^j(x)$ is of the form:

$$\phi_k^j(x) = 2^{\frac{j}{2}} \phi(2^{jx} - k)$$

[34]

According to the preferred embodiment of the invention, it has been observed that the accuracy of the decomposition and the stability of the result of odd degree splines (i.e., linear, cubic, etc.) is improved relative to even degree splines (i.e., quadratic). As such, it is preferred that m, in equation [31] be an odd number (m = 1, 3, 5, . . .).

Upon determining the matrix c^N, process 62 (Figure 10) is then next performed to compute the coefficients of a matrix $\{a\}^N$, using the relationship $a^N = C^N c^N$. According to this preferred embodiment of the invention, the matrix C^N is the inner product of the cardinal spline function $\phi(x)$:

$$C^N = (\langle \phi_k^j, \phi_l^j \rangle)_{k,l=-3}^{2^j-1}$$

[35]

The calculation of process 62 returns the matrix a^N, as shown in Figure 11.

Process 64 is then performed in order to compute intermediate decomposition coefficients utilizing the spline and boundary-wavelet concepts. These intermediate decomposition coefficients are also obtained by way of matrix algebra using digital matrix processor 30 of compressor 16, shown in Figure 5. For the low-frequency, low-pass, or spline component, the intermediate decomposition coefficients a^{N-1} are calculated using the following matrix operation:

$$a^{N-1} = (P^{N-1})^T a^N \qquad [36]$$

where P^{N-1} is a known matrix that contains coefficients of the B-spline to be applied to the input image block, and which has the following form:

$$P^N = \{p^N_{\hat{k},k}\}_{\hat{k}=-m+1,\,k=-m+1}^{2^{j+1},\,2^{j-1}} \qquad [37]$$

The elements p of the matrix of equation [37] are defined as follows:

$$p^N_{k,l} = \frac{B^o_{m,l}\left(\frac{k+m-1}{2}\right) - \sum_{i=k+1}^{i_1} p^N_{i,l} B^o_{m,i}(k+m-1)}{B^o_{m,k}(k+m-1)} \qquad [38]$$

for l from -m+1 to -1, and for k from m+2l to -m+1, where $B^o_{m,i}$ represents the B-spline of order m, on the zeroth level and with ith initial knot.

For the high-frequency, band-pass, or wavelet component of the row of the image block 51, the intermediate decomposition coefficients b^{N-1} are calculated using the following matrix operation:

$$b^{N-1} = (Q^{N-1})^T a^N$$

[39]

where Q^{N-1} is a known matrix containing coefficients of the boundary-wavelets to be applied to the input image block, and having the form:

$$Q^N = \{q^N_{\hat{k},k}\}_{\hat{k}=-m+1,\,k=-m+1}^{2^{j+1}-1,\,2^j-m}$$

[40]

where the elements q of the matrix are defined as follows:

$$q^N_{k,l} = \frac{\sum_{i=i_3}^{i_4} \alpha_{l,i} B^{0,(m)}_{2m,i}(k+m-1) - \sum_{i=k+1}^{i_2} q^j_{i,l} B^o_{m,i}(k+m-1)}{B^o_{m,k}(k+m-1)}$$

[41]

for l from -m+1 to -1, and for k from 3m-2+2l to -m+1, and where $B^o_{m,i}$ represents the B-spline of order m on the zeroth level and with ith initial knot. In equation [41], the values i_2, i_3, i_4 are defined as follows:

i_2 = min (3m-2+2l, k+m-2)
i_3 = max (-m+1, k-m)
i_4 = min (2(l+m-1), k+m-1)

By way of explanation, the sum of the α terms in equation [41] corresponds to the boundary wavelets for those points in image block 51 that are near the edges, and the sum of the q terms in equation [41] corresponds to the inner wavelets for those points in image block 51 that are away from the edges.

According to the preferred embodiment of this invention, each of the matrices C^N, P^{N-1} and Q^{N-1} are pre-calculated and stored in program data memory 35 in each channel compression subsystem 29, in a manner accessible by digital matrix processor 30, so that the matrix operations 62, 64 of Figure 10 can be readily and rapidly performed. The Appendix to this specification specifies examples of these matrices C^N, P^{N-1} and Q^{N-1} which have actually been used for decomposition of an image block of a size 256 pixels square, and where the numeric representation of matrices C, P and Q in the Appendix are given for the cubic (m=4) case.

Upon obtaining the intermediate decomposition coefficients in process 64, the final decomposition coefficient matrices c^{N-1} and d^{N-1}, for the low-pass (spline) and band-pass (wavelet) components, respectively, are obtained by processes 66, 68, respectively. According to this preferred embodiment of the invention which utilizes matrix algebra, process 66 obtains the final low-pass decomposition coefficient matrix c^{N-1} from the relationship:

$$a^{N-1} = C^{N-1}c^{N-1} \qquad [42]$$

??

where the matrix C^{N-1} is the inner product of the cardinal spline function $\phi(x)$, calculated according to equation [35] noted above for the next matrix in sequence. The values of

the matrix cN-1 correspond to the values in the low-frequency portion of processed image block 53 of Figure 9b, and thus represent the function $f_{N-1}(x,y)$ in the spatial domain. Matrix c^{N-1} therefore also represents the frequency domain representation $\hat{f}_N$-1 shown in Figure 8.

Similarly, process 68 of Figure 10 determines the final decomposition coefficient matrix d^{N-1} for the band-pass component of the image block 51. Process 68 is based upon the operation:

$$b^{N-1} = D^{N-1}d^{N-1} \qquad [43]$$

where matrix D^{N-1} is a known matrix that may be precalculated according to the relationship:

$$D^{N-1} = (Q^N)^T C^{N+1}(Q^N) \qquad [44]$$

and is thus based upon known relationships. Upon completion of process 68, the matrix d^{N-1} corresponds to the high-frequency values in processed image block 53 of Figure 9b. The matrix d^{N-1} thus also represents the spatial function $g_{N-1}(x,y)$, and its corresponding frequency domain representation $\hat{g}_N$, which is the band-pass component shown in Figure 8.

Upon completion of the horizontal decomposition of process 50, matrices c^{N-1} and d^{N-1} are stored in memory banks 36 of channel decomposition subsystem 29. preferably in the order illustrated in Figure 9b as processed image block 53. Referring back to Figure 7, process 52 is then performed to again decompose processed image block 53 in similar manner as process 50, only in a vertical direction, i.e., for each vertical column of processed image block 53, taken column-by-column. In other words, the decomposition

of process 52 is performed considering the image data for each vertical column of processed image block 53 as a one-dimensional spatial function in the y-dimension. According to the preferred embodiment of the invention, the matrix algebra used in process 52 is identical to that shown in Figures 10 and 11, for the next level of matrices in sequence, except on a column-by-column basis for each vertical column in processed image block 53.

Process 52 thus decomposes the low-frequency spatial function $f_{N-1}(x,y)$ into a low-frequency portion "LL" and a higher-frequency portion "LH" contained within processed image block 55 as shown in Figure 9c, where the second decomposition is performed on processed image block 53 in a vertical direction. Figure 8 illustrates, in the frequency domain, that the matrix portion LL corresponds to the function $\hat{f}_N\text{-}2(\hat{f}_N\text{-}1)$ that is the low-pass portion of the prior low-pass function $\hat{f}_N\text{-}1$, and that the matrix portion HL corresponds to the function $\hat{g}_N\text{-}1\hat{f}_N\text{-}1$ that is the band-pass portion of the prior low-pass function $\hat{f}_N\text{-}1$.

Similarly, process 52 decomposes the prior band-pass decomposition function $\hat{g}_N\text{-}1$ into a low-frequency (i.e., lower band-pass) portion ("HL" in Figure 9c) and a high-frequency portion ("HH" in Figure 9c), but where the second decomposition is performed on processed image block 53 in a vertical direction. Referring to Figure 8, this decomposition provides frequency domain representation $\hat{f}_N\text{-}2\hat{g}_N\text{-}1$ corresponding to the HL portion of final processed image block 55, and frequency domain representation $\hat{g}_N\text{-}2\hat{g}_N\text{-}1$ corresponding to the HH portion of final processed image block 55.

The LL portion of final processed image block 55 corresponds to the "blur" image of the input video image

block 51, and the LH, HL and HH portions of final processed image block 55 correspond to band-pass portions of the input video image block 51, taken in both the horizontal and vertical directions, successively. As is evident from the frequency domain representations of Figure 8, the LH, HL and HH components correspond to the components of the image in increasing frequency bands, with the HH component corresponding to the highest frequency component.

The decomposition performed in processes 50, 52 described hereinabove provide important benefits in the field of video image decomposition. Firstly, the benefits of time-frequency localization of the signal in a manner in which the frequency-width of the window widens and the time-width of the window shrinks with increasing frequency provided by wavelet decomposition are obtained. This allows for more detailed and thorough analysis of rapidly changing portions of the signal, and thus a more accurate decomposition of the video image. Secondly, by using the boundary-spline-wavelet approach described hereinabove, boundary effects at the edges of the image or of subdivided image blocks within the image are eliminated. The elimination of these boundary effects allows for smaller capacity computers to perform the decomposition with no degradation in image quality. Elimination of boundary effects also enables each subdivided image to be processed independently by a stand-alone device or processor system, so that the entire decomposition may be performed by parallel processing or parallel computing, without extensive modifications. In addition, the decomposed image may, upon decompression, be magnified without the presence of boundary effect artifacts in the displayed image.

Numerical decomposition example

The decomposition of processes 50, 52 can be illustrated by way of a simple numerical example. In this example, the input image block is eight-by-eight pixels in size, and contains an eight-bit digital value representative of the display intensity at that location (i.e., is in PGM format). An example of the input image block in this format is as follows:

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Process 50, as noted above, performs a boundary-spline-wavelet decomposition of the input image block, taken for each horizontal row. In this example, where a trivial zero degree (m=0) wavelet is used, for purposes of explanation, the horizontal decomposition of process 50 results in the matrix:

141.5	151	155	155	|	-2.5	-2	0	0
147.5	154.5	157.5	156	|	-3.5	-1.5	1.5	0
152.5	161.5	157	156	|	-2.5	-1.5	1	0
160	161	159.5	159	|	-1	1	0.5	0
159.5	161.5	158.5	155	|	-0.5	-0.5	3.5	0
161	161	158.5	157	|	0	0	1.5	0
162	162	159.5	157	|	0	1.5	2.5	0
162	161	160.5	158	|	0	0	2.5	0

The portion of the above matrix on the left-hand side of the divider corresponds to the low-frequency portion of the input image, while the portion of the above matrix on the right-hand side of the divider corresponds to the high-frequency portion of the input image, both performed in a horizontal manner.

Process 52 then decomposes the result of process 50, considering the data in vertical columns as one-dimensional spatial sampled functions. Again using the same trivial zero-degree wavelet, the vertical decomposition of process 52 provides the result:

144.5	152.75	156.25	155.5	-3	-1.75	0.75	0
156.25	161.25	158.25	157.5	-1.75	-0.25	0.75	0
160.25	161.25	158.5	156	-0.25	-0.25	2.5	0
162	161.5	160	157.5	0	0.75	2.5	0
-3	-1.75	-1.25	-0.5	0.5	-0.25	-0.75	0
-3.75	-1.75	-1.25	-1.5	-0.75	-1.25	0.25	0
-0.75	0.25	0	-1	-0.25	-0.25	1	0
0	0.5	-0.5	-0.5	0	0.75	0	0

The orientation of the above matrix uses the nomenclature of Figure 9c, as follows:

LL	HL
LH	HH

Video image decomposition example

Referring now to Figures 12a through 12e, an example of the boundary-spline-wavelet decomposition according to the preferred embodiment of the invention as applied to an actual video image will be described. Figure 12a illustrates an input video image in PGM format to which boundary-spline-wavelet decomposition is applied. Figures 12b through 12e illustrate the LL, LH, HL and HH components of the final processed video image after performing processes 50, 52 of Figure 7, using the matrix operations described hereinabove relative to Figures 10 and 11. As is evident from Figure 12b, the LL component is a quite faithful representation of the input image of Figure 12a. In the LH, HL, HH components illustrated in Figures 12c through 12e, respectively, the visible striations correspond to higher frequency components of the input image.

b. Thresholding and compression

Referring back to Figure 6, upon the completion of the horizontal and vertical decomposition of the input image block performed in process 40, process 42 is next performed by quantization processor 32 upon the results of the decomposition stored in memory banks 36.

Specifically, the LH, HL and HH components of the decomposition are subjected to thresholding and quantization in process 42, with no such quantization performed upon component LL. This selection of the LH, HL and HH components for quantization is based upon the observation that most real-world images in PGM format will consist of low-frequency intensities, and that the higher-

frequency wavelet portions of the decomposition will tend to have a large number of small or zero coefficients. The numerical example set forth above relative to the decomposition of the eight-by-eight image illustrates this effect. In addition, the exemplary image of Figures 12a through 12e also shows that the LL component of the decomposition contains the most information, with the LH, HL and HH components containing only a small amount of significant information indicated by the striations therein.

According to the preferred embodiment of the invention, therefore, this large population of small values in the higher-frequency components after decomposition may be either discarded (or reduced in memory requirements) by the quantization of process 42. For example, with reference to the above numerical example, all coefficients having an absolute value less than 2.0 may be set to zero (thresholding), and all remaining coefficients may be rounded to their nearest signed integer value. The memory requirements for storage of the coefficients that undergo the thresholding and quantization are thus much reduced, even before the application of lossless compression techniques as will be noted below.

Other thresholding and quantization techniques may alternatively be applied to the decomposed image data, selectable according to the code stored in quantization data register 31 described hereinabove.

Video image quantization example

Referring now to Figures 13a through 13c, an illustration of the result of thresholding performed in process 42 is illustrated, relative to the video image decompression example of Figures 12a through 12e. Figure 13a is the same image as the LL "blur" component also shown in Figure 12b. Figure 13b illustrates, for example, the sum of the LH, HL and HH coefficients after the decomposition of process 40 of Figure 7, and thus is the sum of the images of Figures 12c through 12e discussed above. Figure 13c illustrates the sum image of Figure 13b after the application of a threshold limit and quantization of process 42. As is evident from Figure 13c, the higher frequency components LH, HL, HH from the decomposition of process 40 correspond only to the edges and other sharp transition locations of the input image.

c. Completion of video compression

Referring back to Figure 6, after the thresholding and quantization of the LH, HL, HH components of the decomposed input image performed in process 42, decision 43 determines whether the desired compression ratio has yet been reached.

As illustrated particularly by the video and numerical examples noted above, upon the completion of a single decomposition process 40, the LL component of the image may be adequate to accurately convey the input image. This single decomposition of the image will provide up to a 4:1 compression ratio, depending upon the memory requirements for the quantized high frequency results. This maximum ratio considers that the wholesale discarding of the higher

frequency components will leave only the LL component residing in a matrix that is one-fourth the size of the input image block. Specifically, therefore, decision 43 determines if the compression ratio achieved so far is adequate for the desired transmission or storage and, if not, passes control back to the decomposition process so that the LL component resulting from the prior decomposition may again be decomposed into four more components.

According to this embodiment of the invention, the determination of whether the desired compression ratio has been obtained may be done relative to a predetermined compression ratio. In such a case, timing circuit 37 will maintain a count of the number of passes through decomposition process 40, and will perform decision 43 by comparing the resulting compression ratio against a previously stored value.

Alternatively, the compression ratio and thus decision 43 may be determined in a dynamic manner, frame by frame, depending upon the accuracy with which the LL component is representative of the input frame image. Conceptually, such a determination will be a measure of the difference between the LL component and the input image block relative to a predetermined accuracy limit, such that if the LL decomposition is within a preselected ϵ limit, an additional pass through the decomposition process 40 may be performed. It is contemplated that this determination can be made automatically by main controller 28 in decompressor 16 so that the process performed in each of channel compression subsystems may remain consistent, by calculating a numerical value based upon the sum of the coefficients in the LH, HL, and HH components of the decomposed image, such sum indicating the difference

between the input image to that pass of the decomposition process and the resultant LL component.

In the usual case, where each video image frame is subdivided into smaller image blocks for compression, it is preferred that the compression ratio be constant for all compressed image blocks of the frame. Accordingly, decision 43 is preferably performed after each image block has been decomposed and quantized in processes 40, 42, to allow the dynamic determination of whether to repeat the decomposition process 40 is made based on the worst case decomposed image.

According to this dynamic determination of decision 43, video image frames that do not contain high frequency information to any large extent, such as background images with smooth transitions between colors, may be compressed to a significantly higher ratio than can video image frames that do contain high-frequency information, or a large number of sharp transitions. Use of such dynamic compression can optimize the overall compression efficiency with minimal impact on the image quality.

Whether statically or dynamically determined, upon decision 43 returning the result that the desired compression ratio has been obtained, according to the preferred embodiment of the invention, lossless compression is then performed upon the results of the decomposed and quantized images for the frame, in process 44 of Figure 6. Referring back to Figure 5, the lossless compression of process 42 is preferably performed by lossless compressor 34 according to a conventional lossless technique such as Huffman encoding. The lossless compression of process 42 is especially beneficial in compressing the quantized higher frequency LH, HL, HH components from the

decomposition, considering that non-zero or varying values in these components will be quite sparse for most video image frames.

After the lossless compression of process 42, the compressed video image data is formatted, or coded, on a frame-by-frame basis for transmission or storage, as the case may be. As noted above, it is contemplated that most video image frames will be compressed according to the preferred embodiment of the invention after subdivision into multiple image blocks, such that a complete frame will consist of multiple ones of such blocks, transmitted together. As illustrated in Figure 4a, it is preferred that the coding of the compressed image data be performed within compressor 16, preferably by data flow interface 39 shown in Figure 5, prior to its application to digital communications network 15.

Referring now to Figure 14, a schematic illustration of formatted frame 70 of data according to the preferred embodiment of the invention will now be described. As will be described in further detail hereinbelow, the decompression of video image data compressed according to the present invention can occur very rapidly; for example, portions of motion pictures compressed according to the present invention have been decompressed and displayed on a real-time basis. This rapid and accurate decompression allows for the ability of enhanced features to be used in the display of the decompressed images. The construction of data frame 70 of Figure 14 includes the compressed bit stream data corresponding to a video frame, plus the appropriate formatting "header" information to allow the enhanced features to be operable. The header portion of frame 70 preferably provides complete information to describe the frame and its position.

As illustrated in Figure 14, frame 70 is a sequential block of data formatted for a storage device such as computer memory, disk storage, CD-ROM and the like. It is therefore contemplated that, in order to take advantage of all of the features of the present invention, compressed data frames 70 will be stored in a computer memory prior to its decompression, rather than decompressed in a real-time manner as received over digital network 15. Real-time decompression and display may alternatively be performed as the data is received, but certain ones of the features described hereinbelow will not be as useful in such a case. Since the extraction of the header information from frame 70 requires extremely little computing time and effort, inclusion of this header information will have substantially no penalty in the overall performance of the real time decompression and display.

The first portion of frame 70 is H1 header 71, which is data of string type used to specify the status of the bit stream compressed image, and as such may contain on the order of twenty bytes of data. For example, H1 header 71 may include an identifier of the movie of which the frame is a part, and may also contain a security code for preventing unauthorized viewing or use of the compressed data. For example, if the compressed video data corresponds to a movie being communicated over digital telephone lines, H1 header may contain a code corresponding to an identifier of a local decompression system 20 so that only that decompression system will be enabled to decompress the data. Field 72 of frame 70 is a four-byte field of long integer type which contains the number of its corresponding frame 70 to specify the position of the frame 20 in the entire video image sequence. Field 73 is a two-byte field of integer type which identifies the group of motion (if any) to which frame 70 belongs. A group of

frames 70 may be designated as a group, such that the frames in the group cannot be separated from one another or cut from the entire sequence. Such grouping can prevent unexpected side effects in the display of the motion picture sequence.

Fields 74 and 75 then follow in frame 70 according to this example, to facilitate control of the display of the video sequence containing frame 70. Field 74 is a four-byte field of long integer type which contains the address at which the previous frame in the sequence begins, enabling rapid jumping back to the previous frame as desired. As will be described hereinbelow, a user control interface may be provided with decompressor system 20 to allow interactive control of the display of the video sequence, in which case field 74 will facilitate the skipping and selection of individual frames in reverse order. Similarly, field 75 is a four-byte field of long integer type which contains the address of the next frame in the sequence, allowing rapid skipping of frames 70 in the forward direction during decompression and display.

Field 76 is a two-byte field of integer type that indicates the complexity of the image contained within frame 70, by specification of compression ratio, quality index, or a user-defined specification of the image, such values useful in measuring and controlling the performance of the decompression and display.

Fields 77, 79, 81, 83 indicate certain parameters used in the compression of the video data for frame 70. The use of header information to communicate these parameters allow for decompression system 20 to have selectable capabilities, such that it can decompress video image data that were compressed according to different techniques. In

addition, since each frame 70 includes these fields 77, 79, 81, 83, the provision of these fields according to the preferred embodiment of the invention enables dynamic compression, such that different compression techniques may be used on different frames 70 in the same video sequence. Field 77 is a two-byte field of integer type that specifies the level of decomposition in the compressed data of frame 70, so that decompression system 20 may perform the proper levels of decompression upon receipt. Field 79 is an integer field indicating the type of lossless compression performed by lossless compressor 34, for example:

0	:	no lossless compression
1	:	Huffman coding
2	:	adaptive Huffman coding

and so on. Field 81 is a two-byte field of integer type that indicates the quantization mode used by quantization processor 32 described hereinabove relative to compression system 10, for example following the tabular listing of the codes noted above. Field 83 is a two-byte field of integer type that stores the quantization value used by quantization processor 32, and thus depends upon the mode indicated in field 81. Using the tabular example noted hereinabove for control of quantization processor 32, the value of field 83 will indicate the quantization value as follows:

Mode # (field 81)	Field 83 represents:
0	0 (don't care)
1	thresholding value
2	dequantizing scalar
3	address of the JPEG quantization table, as an offset from the compressed data
4	address of vector quantization table, as an offset from the compressed data

As indicated in Figure 14, portion 78 of frame 70 contains the compressed image data for field 70, and follows fields 87, 89. Field 87 is a four-byte field of long integer type that specifies the length of portion 78 containing the compressed data stream, and is useful in assisting the reading of the compressed data stream, such as by way of a DMA operation. Field 89 is a four-byte field of long integer type that indicates the offset distance between field 87 and the start of portion 78 containing the compressed image data; field 89 may thus be used to reserve space within frame 70 for other fields and information defined by the user or other developers.

Portion 78 contains a data stream of the compressed video image data for frame 70, as noted above. The length of portion 78 will, of course, depend upon the compression ratio obtained in the compression process, upon the number of pixels in the frame, and also upon the extent to which the lossless compression of process 44 compressed the higher frequency components of the frame. In addition, it is contemplated that audio information may also be encoded in a conventional manner, and placed within portion 78 of

frame 70, to support the transmission or storage of sound motion pictures.

This arrangement of frame 70 is particularly useful in the interactive decompression and display of a sequence of video frames. Specifically, fields 74, 75 and 76 enable decompressor 20 to flexibly display the frames in the sequence, especially in the case where the sequence of frames 70 are sequential frames in a motion picture. For example, decompressor 20 can interrogate field 76 to determine if the processing capacity of decompressor 20 and its display system 26 is such that every frame in the sequence cannot be decompressed and displayed in real time; if so, decompressor 20 can skip to the next frame 70 in the sequence indicated by the contents of field 75 in frame 70. While the quality of the displayed motion picture will be reduced from the best possible images when frames are skipped, those frames that are not skipped are displayed in real-time, so that the time-dependence of the motion in the motion picture is accurately conveyed.

Fields 74, 75, 76 also provide interactive display capability. As field 74 indicates the address of the previous frame in the sequence, frame 70 allows the capability of backwards display of a motion picture, whether for each frame 70 in the sequence or with frames skipped as described above based on the time required for decompression and the capability of decompressor 18. In addition, the information provided by field 76 facilitates the synchronization of the display of the sequence of frames 70, and also allows for easy scaling of the time base to provide slow-motion or enhanced-speed display.

Accordingly, referring back to Figure 6, the formatting or coding of process 46 thus prepares the

compressed video image data for transmission over digital network 15 for storage, decompression and display by decompression system 20, or alternatively to disk unit 22 for archival storage.

4. Construction of the decompressor

a. High performance decompressor

Referring now to Figure 15a, the construction of decompressor 18 according to the preferred embodiment of the invention will now be described in detail. It is, of course, contemplated that other architectural arrangements of circuitry may be used to decompress video data according to the present invention. Specifically, it is contemplated that a conventional general purpose computer system may be capable of performing the compression of the video image data according to the present invention. However, the example of Figure 15a incorporates a preferred embodiment of the circuitry for performing the data decompression functions described herein.

In this example, decompressor 18 preferably includes a data flow interface 80 coupled to network 15 to receive the incoming bitstream data of the compressed video image, as described above relative to Figure 14. Data flow interface 80 provides an interface to network 15, and serves to separate the R, G, and B components of the incoming bitstream into three channels, namely the R-stream, G-stream and B-stream channels. Decompressor 18 also includes main controller 84 and three substantially identical channel decompressor subsystems 88_R, 88_G, 88_B. Main controller 84 preferably controls the operation of the functional circuitry within decompressor 18, and as such is

connected by way of the appropriate bidirectional buses and control lines to subsystems 88 and data flow interface 80 to control the timing, feedback and receiving operation of decompressor 18. Main controller 84 is preferably a general purpose programmable microprocessor or other central processing unit (CPU) of sufficient computational power and capacity to process some or all of the image data and to control the performing of the image compression functions to be described hereinbelow. It is contemplated that microprocessors having performance levels similar to or greater than those of the 80486 type (including those available from Intel Corporation or Cyrix Corporation), of the 68040 type (including those available from Motorola), and of the SPARC processor type (available from Texas Instruments Incorporated or Sun Microsystems, Inc.) will be suitable for use as main controller 84 in decompressor 18.

The three channel compressor subsystems 88_R, 88_G, 88_B each receive a corresponding one of the separated channels presented by data flow interface 80, so that the boundary-spline-wavelet decompression of the three color components is performed in parallel according to this embodiment of the invention. The construction of channel decompressor subsystems 88 is substantially the reverse of that of channel compressor subsystems 29 described hereinabove. Each of subsystems 88_R, 88_G, 88_B is therefore constructed substantially identical to perform these similar tasks as the others, in parallel; for purposes of clarity, the following description is provided only for channel decompressor subsystem 88_R, it being understood that the construction and operation of the other subsystems 88 will be similar, if not identical, to subsystem 88_R described herein.

Each channel decompressor subsystem 88 according to this embodiment of the invention is specifically designed to perform the functions of lossless decompression, dequantization, and boundary spline wavelet re-composition (or decompression), as used in the decompression operation according to this embodiment of the invention. Channel decompressor subsystem 88 according to this embodiment of the invention includes the circuit functions of digital matrix processor 86, timing circuit 107, dequantization processor 108, lossless decompressor 82, program data memory 112, and multiple memory banks 110 for storage of image data.

The incoming data received by subsystem 88, as shown in Figure 15a, is received by lossless decompressor 82. Lossless compressor 82 may be implemented by way of a conventional digital signal processor such as the TMSC40, programmed in such a manner as to perform lossless decompression upon the incoming data stream for that channel as presented by data flow interface 80. The lossless decompression performed by lossless decompressor 82 may be according to a conventional technique, such as Huffman encoding, adaptive Huffman encoding, arithmetic encoding, LSQ coding, and the like; specifically, however, lossless decompressor 80 must either recognize (from field 79 of frame 70) or be previously programmed to operate according to the lossless compression technique used in compression of the incoming data. Lossless decompressor 82 stores the decompressed incoming image data in memory banks 110, in preparation for the re-composition of the image data. Alternatively, lossless decompressor 82 may be implemented as a custom logic circuit for providing this function.

The output from lossless decompressor 82 corresponds to the decomposed image data for that channel (R, G, B), in identical form as that presented to lossless compressor 34 in compressor 16, as described hereinabove. Dequantization processor 108 is a logic circuit or programmable device such as a microprocessor, for de-quantize the output from lossless decompressor 82. According to this preferred embodiment of the invention, dequantization processor 108 is controllable to operate according to various quantization or thresholding modes, according to the contents of dequantization code register 109 which receives the contents of field 81 of frame 70. The dequantization value in field 83 is used in the dequantization performed by dequantization processor 108. The results of the dequantization are stored in memory banks 110.

Digital matrix processor 86 performs the boundary-spline-wavelet composition upon the results of the dequantization, using pre-calculated matrices stored in program data 112. The procedure for this composition is described in detail hereinbelow. The results of the operations performed by digital matrix processor 86 are stored in memory banks 110. As in the case of compressor 16, digital matrix processor 86 may be implemented by way of a conventional digital signal processor, such as the i860 microprocessor available from Intel Corporation and the TMSC40 digital signal processor available from Texas Instruments Incorporated, or by way of a general purpose microprocessor such as those of the 80386 and 80486 families available from Intel Corporation, and of the 68030 and 68040 families available from Motorola.

Timing circuit 107 performs the functions of controlling the storage of the channel data from lossless decompressor 82, and the subsequent retrieval thereof for

dequantization and decompression processes. Upon completion of the decompression of the channel data for a frame, timing circuit 107 is operable to present the decompressed PGM channel data to main controller 84, for formatting and transmission to format converter 24 for display at display 26.

Main controller 84 receives the decompressed channel data from each of subsystems 88_R, 88_G, 88_B and sequences the data into the format suitable for receipt by format converter 24. In addition, main controller 84 operates to provide feedback to data flow interface 80, such that upon decompression of a frame of data, data flow interface 80 can be controlled to communicate the next frame of data to subsystems 88 for decompression.

In addition, main controller 84 is coupled to user interface device 85, which may be a mouse, trackball, pen input, keyboard or the like; in addition, user interface device 85 may be implemented as another computer, such as a personal computer workstation, through which a user controls the decompression and display of video data. As noted hereinabove, the construction of frame 70 shown in Figure 14 facilitates the interactive control of the decompression process, such that the user may control the direction (forward/backward) and rate of decompression and display of the sequence of video image frames, using user interface device 85. Main controller 84 is thus operable to communicate to data flow interface 80 the order and selection of frames for decompression; it is therefore useful to incorporate adequate memory for the storage of multiple frames either within or accessible by data flow interface 80 to allow such interactive decompression and display.

This example of decompressor 18 is intended to support the compression of high definition real-time true color video image data, where "true color" indicates the use of twenty-four bits of color information for each pixel, resulting in 16.7 million possible colors. The decompression rate for decompressor 18 is intended to be on the order of thirty frames per second, so as to support "real-time" video image decompression and display. As noted above, if the color and frame rate requirements are reduced from real-time true color video, it may be possible to implement decompressor 18 as a single channel, i.e. with a single channel compression subsystem 88. In this implementation, color data could be decompressed sequentially for the R, G and B components under the control of main controller 84. In addition, if the frame rate permits, digital matrix processor 86 may be used to perform the lossless decompression, as well.

The construction of decompressor 18 described hereinabove is intended as a "high-end" product utilizing, in a parallel fashion, three processing boards for the three R, G, B channels. This design is preferably used in association with modern high-performance workstations such as the 4D series workstations available from Silicon Graphics, and SPARC workstations available from Sun Microsystems, as these workstations have performance levels adequate for delivering more than thirty frames per second of high resolution (800 pixels by 600 pixels) image data with 24-bit true color capability.

b. Personal-computer expansion card decompression system

Referring now to Figure 19, decompression system 200 according to an alternative embodiment of the invention will now be described in detail. Decompression system 200 is intended for implementation on a conventional PC-based workstation, and is configured as a single 8-bit expansion board; alternatively, system 200 may be configured as a 16-bit ISA or EISA expansion board, or as a MCA (Micro Channel Architecture) board. In each case, it is contemplated that decompression system 200 will be suitable for medium resolution displays.

Decompression system 200 includes lossless decompressor 202, which is implemented as a video processing board. Lossless decompressor 202 has an input port coupled to network 15 (or, of course, to disk storage or another source of video image data compressed according to the method described hereinabove). Lossless decompressor 202 includes digital signal processor (DSP) 204, which is a conventional digital signal processor such as the TMSC25 and TMSC30 types available from Texas Instruments Incorporated, or a general purpose microprocessor such as the 68020 or 68030 type available from Motorola. Program memory 206 is preferably EPROM or other conventional firmware storage, for storing the programs utilized by DSP 204 in performing the lossless decompression upon the received data.

Lossless decompressor 202 receives the compressed RGB-PGM data from network 15 and performs the necessary lossless decompression according to the lossless compression technique used (if any) in compressor 16. It is preferred that lossless decompressor 202 be capable of

performing decompression of various types, selectable by way of field 79 in frame 70, described hereinabove relative to Figure 14. Lossless decompressor 202 is also coupled to user interface device 205, to allow a human user the capability of controlling motion control in the decompression and display of a sequence of video images. Lossless decompressor 202 presents its output to decompression processor 210 for the boundary-spline-wavelet decompression.

Decompression processor 210 according to this embodiment of the invention includes processing capability to perform boundary-spline-wavelet decomposition according to the preferred embodiment of the invention described herein. The main controller of decompression processor 210 is digital signal processor (DSP) 225, which is preferably a high-performance single-chip digital signal processor, such as the TMSC40 available from Texas Instruments Incorporated or the i860 processor available from Intel Corporation, or a high-performance general purpose microprocessor such as the 80486 microprocessor available from Intel Corporation. Decompression processor 210 also includes program data memory 212 for storage of program code and pre-calculated matrices useful in the decompression process, data memory banks 214 for storage of image data, and digital matrix processor 220 for performing matrix operations useful in the decompression routine. Digital matrix processor 220 is preferably a conventional digital signal processor, such as the i860 microprocessor available from Intel Corporation and the TMSC40 digital signal processor available from Texas Instruments Incorporated, or a general purpose microprocessor such as those of the 80386 and 80486 families available from Intel Corporation, and of the 68030 and 68040 families available from Motorola. Host interface controller 208 is also

provided within decompression processor 210, for communicating with a host computer (not shown) including the video display associated therewith. Program memory 222 is also provided in decompression processor 210 and contains the program code for boundary spline wavelet reconstruction and format conversion. Video timing control circuit 217 is also contained within decompression processor 210, for controlling the viewing of the decompressed data as well as the timing of the decompression, so that a sequence of video frames may be viewed in real-time as a movie, or under the control of user interface device 205.

In operation, lossless decompressor 202 stores the results of the lossless decompression for a frame in memory banks 214. These data are retrieved by DSP 225 via memory interface 224, according to a sequence stored in program memory 222, and forwarded to digital matrix processor 220 along with precalculated matrices stored in program data 212 along the coefficient bus shown in Figure 19. Digital matrix processor 220 performs matrix operations upon the image data presented by DSP 225 to perform the boundary-spline-wavelet decomposition and forwards the results to DSP 225 along the data-out bus shown in Figure 19 for storage in memory banks 214. Digital matrix processor 220 also provides control information to DSP 225 on the feedback bus therebetween.

Upon completion of the boundary-spline-wavelet reconstruction or decompression, the decompressed image data for a frame is stored in memory banks 214. Memory interface 224 enables the interleaved or simultaneous accessing of memory banks 214 for boundary-spline-wavelet decompression and display; host interface controller 208 accesses memory banks 214 for display purposes. Host

interface controller 208 may also have sufficient capability to directly display the contents of memory banks 214 containing the decompressed video image data; alternatively, the decompressed image data may be forwarded by host interface controller 208 to memory of the host PC for eventual display, by way of DMA.

It is contemplated, of course, that still further alternatives to the decompression hardware may be utilized according to the present invention.

5. Boundary-spline-wavelet video image data decompression and display

Referring now to Figure 16, a method according to the preferred embodiment of the invention, for decompressing video image data that was compressed by way of boundary-spline-wavelet compression as discussed above, will now be described in detail. This decompression will be described relative to the hardware decompression system of Figures 4a and 15a, it being understood that analogous operations would be carried out by decompression system 200 illustrated in Figure 19, for its particular implementation.

The initial step of the decompression method of this example is the reading of the compressed file or sequence of frames 70 by data flow interface 80, indicated in Figure 16 by process 90. Process 90 may be the result of the receipt of communicated compressed video image data by data flow interface 80 directly from digital network 15. Alternatively, where decompressor 18 is a portion of a stand-alone computer, disk unit 22 or another fixed storage unit may contain the file of compressed video image frames

from an earlier transmission, or as archival storage. In either case, the result of process 90 is the receipt of a sequence of compressed video image frames 70 by data flow interface 80, from which the remainder of the process of Figure 16 can retrieve individual frames 70.

Process 91 is next performed, where each of the channels of a selected frame 70_n undergoes lossless decompression by lossless decompressor 82. As noted above relative to Figure 15a, the channels of data are decompressed in parallel according to the preferred embodiment of the invention, and as such the process flow of Figure 16 from this point forward will be described for a single channel (R, G, B) of frame 70_n. The particular frame 70_n is selected according to the sequence of frames 70 in the motion picture, or is alternatively selected by the user via user interface device 85. In either case, main controller 84 controls the access of selected frame 70_n, so that the appropriate frame 70 is presented to channel decompressor subsystems 88, which provide the data for frame 70_n to lossless decompressor 82. As noted above, the lossless decompression performed in process 91 by lossless decompressor 82 corresponds to the type of lossless compression performed by lossless compressor 34 in compressor 16. The result of the lossless decompression of process 91 thus provides the final decomposed image blocks 55 of frame 70_n, quantized as described hereinabove relative to the compression method.

After the lossless decompression of process 91, the image data is dequantized for each channel R, G, B by dequantization processor 108 in its corresponding channel decompression subsystem 88. The type of dequantization performed by dequantization processor 108 is controlled by the contents of dequantization data register 109, which

contains a value corresponding to that transmitted as field 81 in frame 70_n. In the event that the data was quantized, the quantization value transmitted in field 83 of frame 70_n is utilized to restore the data to that prior to quantization in compressor 16.

Upon completion of the dequantization of process 92, process 93 is performed for each image block of frame 70_n to reconstruct the full video image in PGM format, with each pixel location containing a value corresponding to the intensity at that location. In effect, process 93 reverses the decomposition process described hereinabove relative to Figures 6 and 7, according to the boundary-spline-wavelet approach. This reconstruction of process 93 is performed on a block-by-block basis, for each of the image blocks in the selected frame 70_n. Upon completion of process 93, the selected frame 70_n is stored as a PGM format frame in memory banks 110.

Referring now to Figure 17, a method for reconstructing the selected frame 70_n according to the preferred embodiment of the invention will now be described in further detail. Recalling that the decomposition of the video image data performed during compression is performed first in the horizontal direction and then in the vertical direction, the decomposition must be performed in the reverse order. The first step in the reconstruction of the video image data is therefore the reconstruction of each column in the vertical direction, as indicated by process 98.

The reconstruction process 98 is also most efficiently, and thus preferably, performed by digital matrix processor 86 in the corresponding channel decompression subsystem 88 using matrix algebra operations

similarly as in the compression process described hereinabove. Accordingly, process 98 begins by considering the image block in columns, each containing a set of coefficients c^{N-k} for the low frequency portion and a set of coefficients d^{N-k} for the high frequency portion, with k being the number of times that boundary-spline-wavelet decomposition was performed during compression to obtain the desired compression ratio. For example, where the compression ratio was 4:1, the value of k would equal 2 (i.e., both a row and a column decomposition was performed once); for a compression ratio of 64:1, k would equal 6.

Figure 18 illustrates the relationship of the matrices used in the reconstruction of the image block performed in process 98. Each reconstruction operation (process 98) is directed to obtaining the next higher order coefficient values c_{N-k+1}. Conversely to the case of compression, the matrix operation is equivalent to:

$$c^{N-k+1} = P^{N-k}c^{N-k} + Q^{N-k}d^{N-k} \qquad [45]$$

where the matrices P^N and Q^N are the spline and wavelet matrices, respectively, for level N, as used in the decomposition and compression process. As noted above, matrices P^N and Q^N can be pre-calculated and stored in the program data memory 112 of channel decompression subsystems 88.

Process 98 thus performs one level of the matrix operations of equation [45] on a column-by-column basis for each column in the image block. Upon completion of process 98, process 100 performs a similar reconstruction as that in process 98, only on a row-by-row basis for each row in the image block. As such, the operation of equation [45] is performed to obtain the coefficients c^{N-k+2}.

It will, of course, be noted that the decompression of the video image data according to the preferred embodiment of the invention will not result in an exact duplicate of the input image presented to the compression process, as the effects of the quantization and thresholding process in the compression of the video image data cannot be full recovered (i.e., some data is lost). As such, the boundary-spline-wavelet compression according to the present invention is a lossy compression technique. It is contemplated, however, that the use of the boundary-spline-wavelet approach described hereinabove causes the loss of only that data at high-frequencies, such that the loss of quality of the image is minimized for a given compression ratio.

However, it should especially be noted that the decompression processes 98, 100 can be done quite rapidly, particularly with specialized circuitry such as a digital matrix processor performing the matrix operations noted above. This speed in processing arises from the wavelet forms selected in the compression, which correspond to express mathematical formulae and which are also quite adaptable to matrix operation techniques. As such, it is contemplated that the decompression performed by decompressor 18 can occur quite rapidly and, for many applications, need not require high performance microprocessor circuitry.

Upon the reconstruction in both the column and row directions as performed in processes 98, 100, decision 101 is performed to determine if the image block has been fully reconstructed (i.e., if the image block corresponds to a PGM format image, and does not contain low and high frequency components). If not, the matrix operations of

processes 98, 100 are repeated to perform the next levels of reconstruction of the image for that image block.

Upon process 100 completing the reconstruction of the image block, decision 103 tests whether more image blocks remain to be reconstructed in the selected frame 70_n. If so, the next image block is retrieved from memory 80 (process 104) and is reconstructed by processes 98, 100, 101 as before. Upon completion of the reconstruction of all image blocks in selected frame 70_n, process 106 is performed so that the reconstructed image blocks in frame 70_n are stored in image buffer 88, arranged in the proper order to correspond to a full PGM format image frame, completing process 93 of Figure 16 for a single frame 70_n. The decompressed image data for the frame 70_n, including that for all three channels R, G, B, are the forwarded to main controller 84 for communication to format converter 24. Main controller 84 then indicates completion of a frame to data flow interface and performs decision 95 to determine if additional frames are to be decompressed and display; if so, main controller 84 will select the next frame for decomposition in the sequence, or as directed by user interface device 85.

Referring back to Figure 16, process 94 is then performed by way of which format converter 24 reformats and communicates the frame to display 26, in the desired fashion. It is contemplated that format converter 24 may perform such operations as gathering three successive PGM format frames, corresponding to the red, green, and blue color components of the same image, and presenting the data in RGB-PGM format to display 26; format converter 24 may also convert the RGB-PGM format data into another format, such as PCX, IMG, GIF, TIF, RLE, NTSC and the like, as appropriate for display 26.

Referring now to Figure 15b, the construction of format converter 24 according to the preferred embodiment of the invention will now be described in detail. Format converter 24 includes three buffers 120_R, 120_G, 120_B for receiving from decompressor 18 and storing PGM format data corresponding to the three decompressed R, G, B channels. The outputs of buffers 120 are received by color combiner circuit 122, which is a data processing circuit such as a microprocessor, which combines the decompressed video data for the three R, G, B channels into RGB-PGM format. Color combiner 122 presents the results of this combination to the appropriate formatting circuitry for driving display 26. In the analog case, color combiner 122 presents its output to video digital-to-analog converter (DAC) 126; video DAC 126 is a conventional video DAC suitable for receiving digital video or graphics information and driving analog display 26a. In the digital case, color combiner 122 presents its output to color reducer and format encoder 124. Reducer/encoder 124 is a conventional circuit for encoding the RGB-PGM format data presented thereto by color combiner 122 into the desired format for digital display 26d, such formats including TIF, IMG, GIF and the like. Of course, if display 26 is capable of directly displaying RGB-PGM video data, reducer/encoder 124 is unnecessary.

In addition, format converter 24 may also include graphics processing capability so as to perform more graphics processing operations, including magnification, zooming and the like. As noted above, the elimination of boundary effects resulting from the boundary-spline-wavelet compression and decompression according to the present invention is especially beneficial when such complex graphics processing features. For example, a portion of the image of frame 70_n may be readily magnified according to conventional graphics processing techniques, such as the

use of interpolation techniques to fill in pixels between those for which the frame contains actual data, with a high degree of accuracy. In contrast, boundary effects that result from prior compression techniques tend to be exaggerated when the image is magnified, considering not only the magnified display but also the exaggeration resulting from interpolation between "true" image data and the boundary effect artifacts.

Indeed, it is contemplated that the present invention will be especially beneficial for the transmission and storage of compressed video images for display on advanced displays that have higher pixel densities than the density of the image so compressed, due to the elimination of boundary effect artifacts by the present invention. In addition, it is contemplated that such magnification will allow for the display of the images on a plurality of video displays arranged in an array, with each display showing only a portion of the full image.

Referring back to Figure 16, upon transmission of the selected frame 70_n to format converter 24 performed in process 94, decision 95 is then performed to determine if additional frames in the sequence are to be decompressed for display. If so, process 96 selects the next frame 70 for decompression and display. According to the preferred embodiment of the invention, the selection of process 96 may be made in multiple ways. A first approach for selecting the next frame for decompression and display is merely to select the next frame in the sequence, and access the same by way of field 75 (Figure 14) from the previously decompressed frame 70_n; this approach will be followed during the decompression and display of a motion picture.

As noted above relative to Figure 14, field 76 includes data indicating such factors as the compression ratio and the like that indicate the computational complexity required for the decompression and display of its frame. This allows for intelligent selection of the next frame in process 96, where the capability of decompression system 20 is limited relative to the amount of compressed video data in the transmission. For example, if the time that decompressor 18 will require for the decompression and display and display of a frame is longer than the reciprocal of the frequency at which frames are to be displayed (e.g., longer than 1/30 sec. in the case of a conventional motion picture), main controller 84 will skip the next frame or frames, and select a later frame for decompression and display so that the sequence of frames that are displayed will appear in real-time fashion on display 26.

A third approach for the selection of the next frame in process 96 is that which may be directed by the user via user interface device 85. For example, the viewer may wish to display the sequence on a non-real-time basis to allow for intense study of each frame, in either a forward or backward direction, or the viewer may wish to repetitively view a selected portion of the sequence. These commands are conveyed to main controller 84 by user interface device 85, and are used in process 96 according to the present invention to select the next frame.

Upon the decompression and display of all of the frames in the sequence as determined by decision 95, the process of decompression and display ends.

5. Conclusion

The methods and systems for compressing and decompressing video image data described hereinabove relative to the present invention provide important advantages, as noted throughout the foregoing specification. These advantages include the fundamental benefits of wavelet analysis, where time-frequency localization of the input signal is implemented so that the time window narrows with increasing frequency and widens with decreasing frequency, thus providing highly accurate analysis for transient periods of the signal, and avoiding the undersampling problems of conventional Fourier analysis techniques.

In addition, the video image compression and decompression techniques according to the preferred embodiments of the invention utilize boundary-spline-wavelets to eliminate boundary effects in the compressed images, specifically by using different wavelets for samples near the boundaries of an interval from those wavelets used for inner samples. The boundary wavelets do not require support from outside of the interval; when applied to video compression according to the present invention, therefore, boundary effect artifacts are eliminated. This not only provides an extremely accurate representation of the original input image but, when coupled with the computational geometrical superiority of spline functions, also enables the performance of complex processing operations and display such as magnification, dynamic compression on a frame-by-frame basis, interactive display of a motion image, including insertion, editing and repetitive display. In addition, the present invention also enables slow display systems to skip frames so that a

motion picture can be displayed on a real-time basis with fewer frames per second.

The elimination of boundary effects also enables the compression routine to be readily applied to subdivided blocks of the input image, without sacrificing image quality upon decompression and display. This allows for parallel processing techniques to be readily applied to the compression and decompression operations. In addition, lower-capacity computing equipment is enabled by the present invention to perform compression and decompression, while still obtaining high compression ratios and excellent picture quality.

In addition, the present invention also enables a relatively higher compression ratio than prior systems, for a given accuracy, due to the improved time-frequency localization provided by the selected wavelets according to the present invention, and also the ability to compress to a desired accuracy limit as described above. The present invention further enables the dynamic compression of a sequence of frames, so that high frequency frames may be compressed to different ratios than low frequency frames.

The present invention may be applied to various applications of video image compression, such as the transmission of motion pictures and other video data between remote locations, such as over digital communication networks or by satellite transmission. The present invention may also be applied advantageously to the archival storage of video image data, both motion picture and still image data, in a compressed form. The archival storage of both motion video and still images is especially facilitated by the present invention due to the lack of boundary effects, as the decompression and display of the

compressed information may be performed in much higher density and capability systems than those used in the original compression, without exaggeration of artifacts such as boundary effects.

While the invention has been described herein relative to its preferred embodiments, it is of course contemplated that modifications of, and alternatives to, these embodiments, such modifications and alternatives obtaining the advantages and benefits of this invention, will be apparent to those of ordinary skill in the art having reference to this specification and its drawings. It is contemplated that such modifications and alternatives are within the scope of this invention as subsequently claimed herein.

APPENDIX

$$
(P^N)^T = \frac{1}{32}\begin{pmatrix}
16 & 8 & 0 & \cdots \\
0 & 8 & 12 & 3 & 0 & \cdots \\
0 & 0 & 4 & 11 & 8 & 2 & 0 & \cdots \\
0 & 0 & 0 & 2 & 8 & 12 & 8 & 2 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 2 & 8 & 12 & 2 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 8 & 12 & 8 & 2 & 0 & \cdots \\
\vdots & & & & \vdots & & & \vdots & & & & \vdots \\
0 & & & \cdots & 0 & 2 & 8 & 12 & 8 & 2 & 0 & 0 & 0 \\
0 & & & & & \cdots & 0 & 2 & 8 & 11 & 4 & 0 & 0 \\
0 & & & & & & & \cdots & 0 & 3 & 12 & 8 & 0 \\
0 & & & & & & & & & \cdots & 0 & 8 & 16
\end{pmatrix}
$$

$$
C^N = \frac{1}{81}\begin{pmatrix}
720 & 441 & 93 & 6 & 0 & \cdots \\
441 & 1116 & \frac{1575}{2} & 174 & \frac{3}{2} & 0 & \cdots \\
93 & \frac{1575}{2} & 1647 & 1132 & \frac{239}{2} & 1 & 0 & \cdots \\
6 & 174 & 1132 & 2416 & 1191 & 120 & 1 & 0 & \cdots \\
0 & \frac{3}{2} & \frac{239}{2} & 1191 & 2416 & 1191 & 120 & 1 & 0 & \cdots \\
0 & 0 & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & \cdots \\
0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & \cdots & 0 & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & 0 \\
 & & & \cdots & 0 & 1 & 120 & 1191 & 2416 & 1191 & \frac{239}{2} & \frac{3}{2} & 0 \\
 & & & & \cdots & 0 & 1 & 120 & 1191 & 2416 & 1132 & 174 & 6 \\
 & & & & & \cdots & 0 & 1 & \frac{239}{2} & 1132 & 1647 & \frac{1575}{2} & 93 \\
 & & & & & & \cdots & 0 & \frac{3}{2} & 174 & \frac{1575}{2} & 116 & 441 \\
 & & & & & & & \cdots & 0 & 6 & 93 & 441 & 720
\end{pmatrix}
$$

i	$q_{-3,i}$	$q_{-2,i}$	$q_{-1,i}$	$q_{0,i}$
-3	$-\frac{1136914560}{27877}$	$\frac{2387315060}{195139}$	$-\frac{123066720}{1365937}$	0
-2	$\frac{1655323200}{27877}$	$-\frac{2141121840}{195139}$	$\frac{2226000}{1365937}$	0
-1	$-\frac{1321223960}{27877}$	$-\frac{878161880}{195139}$	$-\frac{188417600}{1365937}$	0
0	$\frac{633094403}{27877}$	$\frac{498772701}{27877}$	$\frac{2293862247}{1365937}$	1
1	$-\frac{229000092}{27877}$	$-\frac{4726413628}{195139}$	$-\frac{10796596516}{1365937}$	- 124
2	$\frac{46819570}{27877}$	$\frac{3606490941}{195139}$	$\frac{25245248833}{1365937}$	167
3	- 124	- 7904	- 24264	- 7904
4	1	1677	18482	18482
5	0	- 124	- 7904	- 2426
6	0	1	1677	18482
7	0	0	- 124	- 790
8	0	0	1	1677
9	0	0	0	- 124
10	0	0	0	1

WE CLAIM:

1. A method of compressing digital data representative of a video image for communication thereof, comprising the steps of:

formatting the digital data corresponding to a frame of the video image into an array of locations in rows and columns, with the value at each array location corresponding to the intensity of the display;

decomposing, in a first direction, each row of said array into low-frequency and high-frequency portions using boundary-spline-wavelets, and arranging the results of the decomposing into corresponding rows;

then decomposing each column of said arranged results of the first decomposing step in a second direction, using boundary-spline-wavelets; and

communicating the results of the decomposing steps to a receiving unit.

2. The method of claim 1, wherein said communicating step comprises:

storing the results of the decomposing steps in a fixed memory.

3. The method of claim 1, wherein said communicating step comprises:

transmitting the results of the decomposing steps to a decompression system for display.

4. The method of claim 3, further comprising:

after receiving the transmitted results of the decomposing steps, reconstructing the transmitted video image in the second direction, using boundary-spline-wavelets;

then reconstructing the results of the first reconstructing step in the first direction, using boundary-spline-wavelets; and

displaying the results of the reconstructing steps on a video display.

5. The method of claim 1, wherein the results of the second decomposing step comprise:

an LL component, corresponding to the low-frequency component, taken in the second direction, of the low-frequency decomposition in the first direction of the first decomposing step;

an LH component, corresponding to the high-frequency component, taken in the second direction, of the low-frequency decomposition in the first direction of the first decomposing step;

an HL component, corresponding to the low-frequency component, taken in the second direction, of the high-frequency decomposition in the first direction of the first decomposing step; and

an HH component, corresponding to the high-frequency component, taken in the second direction, of the high-frequency decomposition in the first direction of the first decomposing step;

and further comprising:

quantizing the LH, HL, and HH components prior to said communicating step.

6. The method of claim 5, further comprising:

after said quantizing step and prior to said communicating step, performing lossless compression on the LL, LH, HL, and HH components.

7. The method of claim 5, further comprising:

repeating said decomposing steps.

8. The method of claim 5, further comprising:

transmitting the LL component and the quantized LH, HL, HH components over a communications network;

receiving the transmitted components from the communications network;

reconstructing a low-frequency component from the LL and quantized LH components using boundary-spline-wavelets in the second direction, and reconstructing a high-frequency component from the quantized HL and HH components using boundary-spline-wavelets in the second direction;

then reconstructing a video image from said low-frequency component and said high-frequency component using boundary-spline-wavelets applied in the first direction; and

displaying the results of the reconstructing steps on a video display.

9. The method of claim 8, further comprising:

after said quantizing step and prior to said communicating step, performing lossless compression on the LL, LH, HL, and HH components; and

after said receiving step, performing lossless decompression on the communicated LL, LH, HL and HH components.

10. The method of claim 8, further comprising:

repeating said decomposing steps a selected number of times prior to said communicating step; and

repeating said reconstructing steps the selected number of times prior to said displaying step.

11. The method of claim 5, further comprising:

prior to said decomposing steps, dividing each frame into a plurality of image blocks;

wherein said decomposing steps are performed for each image block of each frame, so that each of said image blocks comprises a LL component and quantized LH, HL, and HH components.

12. The method of claim 11, further comprising:

transmitting the LL component and the quantized LH, HL, HH components for each of said image blocks of each frame over a communications network;

receiving the transmitted components from the communications network;

for each of said image blocks, reconstructing a low-frequency component from the LL and quantized LH components using boundary-spline-wavelets in the second direction, and reconstructing a high-frequency component from the quantized HL and HH components using boundary-spline-wavelets in the second direction;

then, for each of said image blocks, reconstructing a video image from said low-frequency component and said high-frequency component using boundary-spline-wavelets applied in the first direction;

arranging the reconstructed video image for each of said image blocks into a video frame; and

displaying the video frame on a video display.

13. A method of displaying compressed video image data, comprising:

receiving compressed video image data corresponding to a frame and storing the compressed frame in memory, said compressed video image data arranged as first, second, third and fourth frequency domain components;

in a first image direction, applying boundary-spline-wavelets to each column of data in the compressed frame to reconstruct a low-frequency component from the

first and second frequency domain components, and to reconstruct a high-frequency component from the third and fourth frequency domain components; in a second image direction, applying boundary-spline-wavelets to each row of data of the low-frequency component and high-frequency component to reconstruct the video image frame;

displaying the reconstructed video image frame on a video display.

14. The method of claim 13, further comprising:

repeating said steps of applying boundary-spline-wavelets in the first and second image directions a selected number of times.

15. The method of claim 13, wherein said steps of applying boundary-spline-wavelets are performed, for each frame, for a plurality of image blocks of said frame.

16. The method of claim 13, further comprising:

performing lossless decompression on the compressed video image data, prior to said applying steps.

17. The method of claim 13, further comprising:

after said applying steps, storing the reconstructed video image frame in an image buffer.

18. The method of claim 17, further comprising:

after said storing step, magnifying the image of said video image frame prior to said displaying step.

19. A system for communicating video image information, comprising:

an input source for providing digital video image information arranged as frames; and

compressor circuitry having an input coupled to said input source, for decomposing each frame of digital video image information in a first image direction using a boundary-spline-wavelet into a first low-frequency component and a first high-frequency component, and for then further decomposing each of said low-frequency and high-frequency components of each frame of digital video image information in a second image direction using a boundary-spline-wavelet into first and second pairs of low-frequency and second high-frequency components, said first pair being the decomposed representation of said first low-frequency component and said second pair corresponding to a decomposed representation of said first high-frequency component, said compressor circuitry also having an output for presenting the first and second pairs of low-frequency and high-frequency components.

20. The system of claim 19, further comprising:

a format converter circuit coupled between said input video source and said compressor, for converting the digital input video information into a portable gray-level format.

21. The system of claim 20, wherein said input source provides color video information;

and wherein said format converter circuit is for converting the color video information into a plurality of portable gray-level format representations, each representation corresponding to a color component.

22. The system of claim 19, wherein said compressor circuitry comprises:

decomposing circuitry for performing the decomposing of each frame of digital video image

information into the first and second pairs of low-frequency and high-frequency components;

quantization circuitry for quantizing the high-frequency component of said first pair, and for quantizing the both the low-frequency and the high-frequency component of the second pair; and

lossless compression circuitry, having an input coupled to said quantization circuitry, for performing lossless compression of the first and second low-frequency and high-frequency pairs, prior to presenting the decomposed frames of digital video image information at the output of the compressor circuitry.

23. The system of claim 22, further comprising:

main controller circuitry, for controlling the operation of the decomposing circuitry, so that the low-frequency component of said first pair may be repetitively provided to the decomposing circuitry for further decomposing.

24. The system of claim 19, wherein said compressor circuitry comprises:

an image buffer coupled to the input of the compressor circuitry for storing a frame of digital video image information; and

a digital matrix processor for performing the decomposing of each frame of digital video image information into the first and second pairs of low-frequency and high-frequency components using a boundary-spline-wavelet.

25. The system of claim 24, wherein said digital matrix processor operates according to matrix operations using precalculated matrices corresponding to spline and wavelet components and stored in said compressor circuitry.

26. The system of claim 19, further comprising:

a storage unit, coupled to the output of said compressor circuitry, for storing the decomposed frames of digital video image information.

27. The system of claim 19, wherein the output of said compressor circuitry is coupled to a digital network for communication of said decomposed frames of digital video image information.

28. The system of claim 27, wherein said compressor circuitry comprises:

decomposing circuitry for performing the decomposing of each frame of digital video image information into the first and second pairs of low-frequency and high-frequency components;

quantization circuitry for quantizing the high-frequency component of said first pair, and for quantizing the both the low-frequency and the high-frequency component of the second pair; and

lossless compression circuitry, having an input coupled to said quantization circuitry, for performing lossless compression of the first and second low-frequency and high-frequency pairs, prior to presenting the decomposed frames of digital video image information at the output of the compressor circuitry;

and wherein the output of the lossless compression circuitry is coupled to the digital network so that the decomposed frames of digital video image information are communicated after the lossless compression.

29. The system of claim 19, further comprising:

a communications network coupled to the output of the compressor circuitry;

decompressor circuitry, having an input coupled to said communications network, for reconstructing each decomposed frame of digital video image information in the second image direction using a boundary-spline-wavelet, and for then further reconstructing each frame of decomposed digital video image information in the first image direction using a boundary-spline-wavelet; and

a video display, for displaying the reconstructed frames of digital video image information.

30. The system of claim 29, wherein said compressor circuitry comprises:

decomposing circuitry for performing the decomposing of each frame of digital video image information into the first and second pairs of low-frequency and high-frequency components;

quantization circuitry for quantizing the high-frequency component of said first pair, and for quantizing the both the low-frequency and the high-frequency component of the second pair; and

and wherein said decompressor circuitry comprises:

reconstructing circuitry, for reconstructing a digital output video frame from the decomposed frames of digital video image information, by reconstructing an approximation of the first low-frequency component from the first low-frequency and high-frequency pair in the second image direction, and an approximation of the first high-frequency component from the second low-frequency and high-frequency pair in the second image direction, and by then reconstructing the digital output video frame from the approximations of the first low-frequency component and the first high-frequency component in the first image direction.

31. The system of claim 30, wherein the compressor circuitry further comprises:

main controller circuitry for controlling the operation of the decomposing circuitry, so that the low-frequency component of said first pair may be repetitively provided to the decomposing circuitry for further decomposing.

and wherein the decompressor circuitry further comprises:

main controller circuitry, for controlling the operation of the reconstructing circuitry so that the result of the reconstructing may be repetitively operated upon by the reconstructing circuitry, depending upon the number of times the decomposing circuitry repetitively decomposed the digital input video frame.

32. The system of claim 30, wherein said compressor circuitry further comprises:

lossless compression circuitry, having an input coupled to said quantization circuitry, for performing lossless compression of the first and second low-frequency and high-frequency pairs, prior to presenting the decomposed frames of digital video image information to the communications network;

and wherein said decompressor circuitry further comprises:

lossless decompression circuitry, having an input coupled to the communications network for performing lossless decompression of the digital video image information received therefrom.

33. A system for displaying compressed video image data, comprising:

a memory for storing a plurality of compressed video frames;

decompressor circuitry for reconstructing each of said plurality of compressed frames in a first image direction using a boundary-spline-wavelet, and for then further reconstructing each of said plurality of compressed frames in a second image direction using a boundary-spline-wavelet; and

a video display, coupled to said decompressor circuitry, for displaying each of said plurality of reconstructed frames.

34. The system of claim 33, wherein said memory is coupled to a communications network.

35. The system of claim 33, wherein said decompressor circuitry comprises:

a digital matrix processor coupled to said memory, for reconstructing each of said plurality of frames using precalculated matrices.

36. The system of claim 33, wherein each of said plurality of frames are stored in said memory in a form of the type comprising quantized first and second pairs of low-frequency and high-frequency components, said first pair of low-frequency and high-frequency components corresponding to the decomposition, in the first direction, of a low-frequency decomposition in the second direction of a video image frame, and said second pair of low-frequency and high-frequency components corresponding to the decomposition, in the first direction, of a high-frequency decomposition in the second direction of the video image frame.

37. The system of claim 36, wherein each of said plurality of frames are stored in said memory in a form including the lossless compression of said quantized first

and second pairs of low-frequency and high-frequency components;

and further comprising:

lossless decompression circuitry, coupled between said memory and said decompressor circuitry, for performing lossless decompression of each of said plurality of frames prior to the reconstruction thereof by said decompressor circuitry.

38. The system of claim 36, wherein said first and second pairs of low-frequency and high-frequency components correspond to the results of repetitively decomposed components of a video image;

and further comprising:

main controller circuitry, for controlling the operation of the decompressing circuitry so that the result of the reconstructing may be repetitively operated upon by the decompressing circuitry according to the number of times that the stored frames were repetitively decomposed.

39. The system of claim 33, wherein each of said plurality of frames stored in said memory correspond to a plurality of compressed image blocks;

and wherein said decompressor circuitry reconstructs each of the compressed image blocks for each of said plurality of compressed frames in the first image direction using a boundary-spline-wavelet, and for then further reconstructing each of the compressed image blocks for said plurality of compressed frames in the second image direction using a boundary-spline-wavelet.

* * * * *

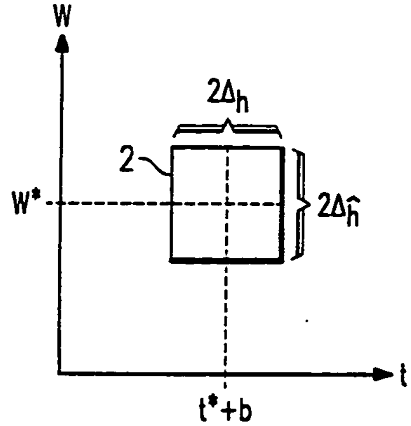


FIG. 1a

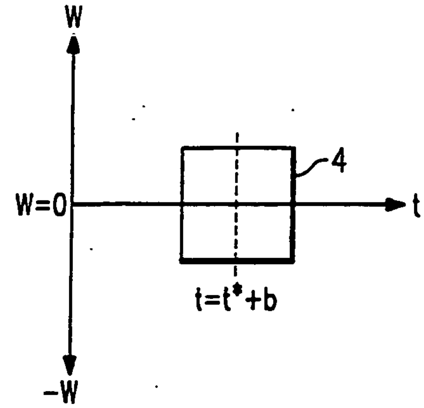


FIG. 1b

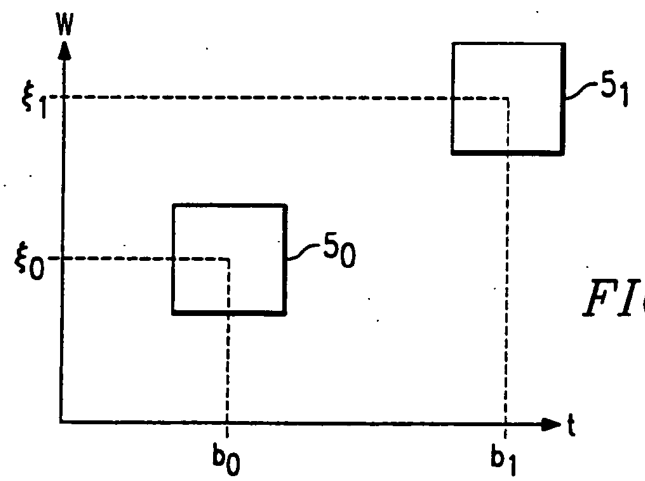
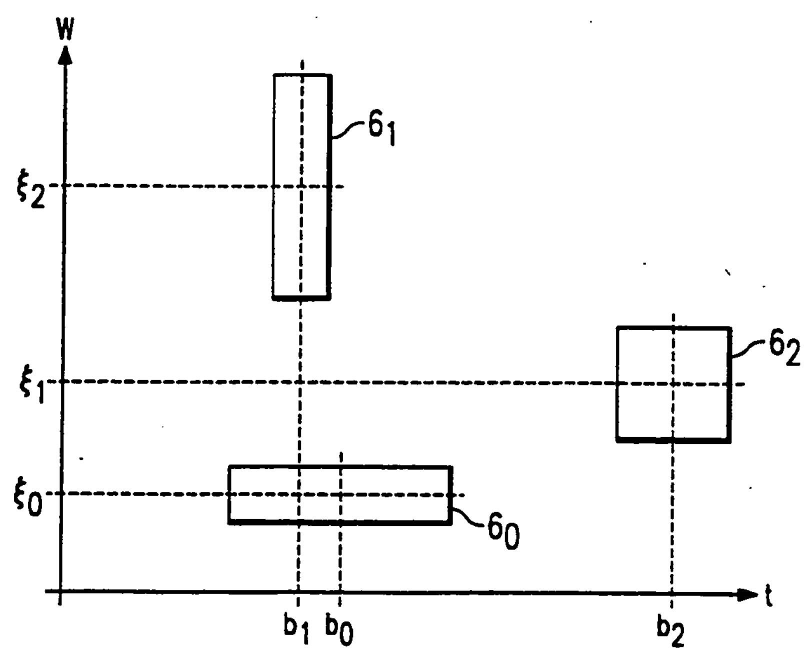


FIG. 1c

FIG. 1d



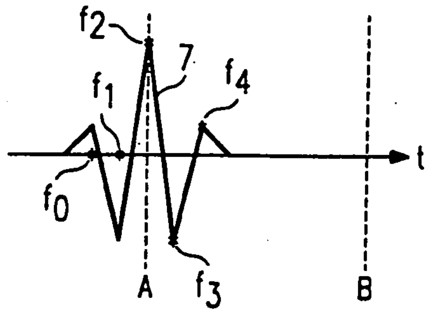


FIG. 2

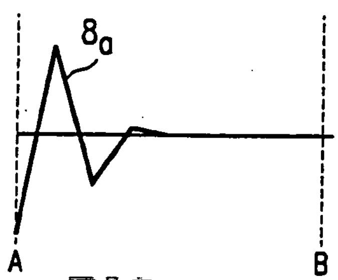


FIG. 3a

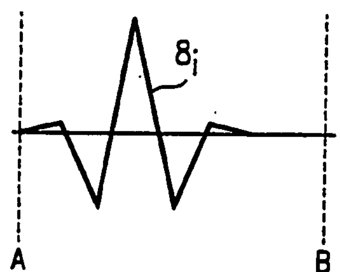


FIG. 3b

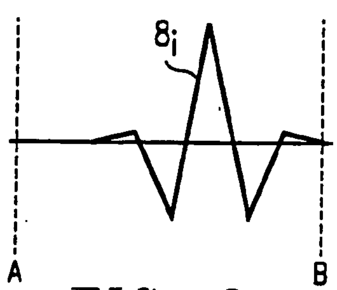


FIG. 3c

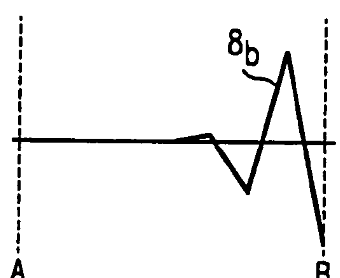


FIG. 3d

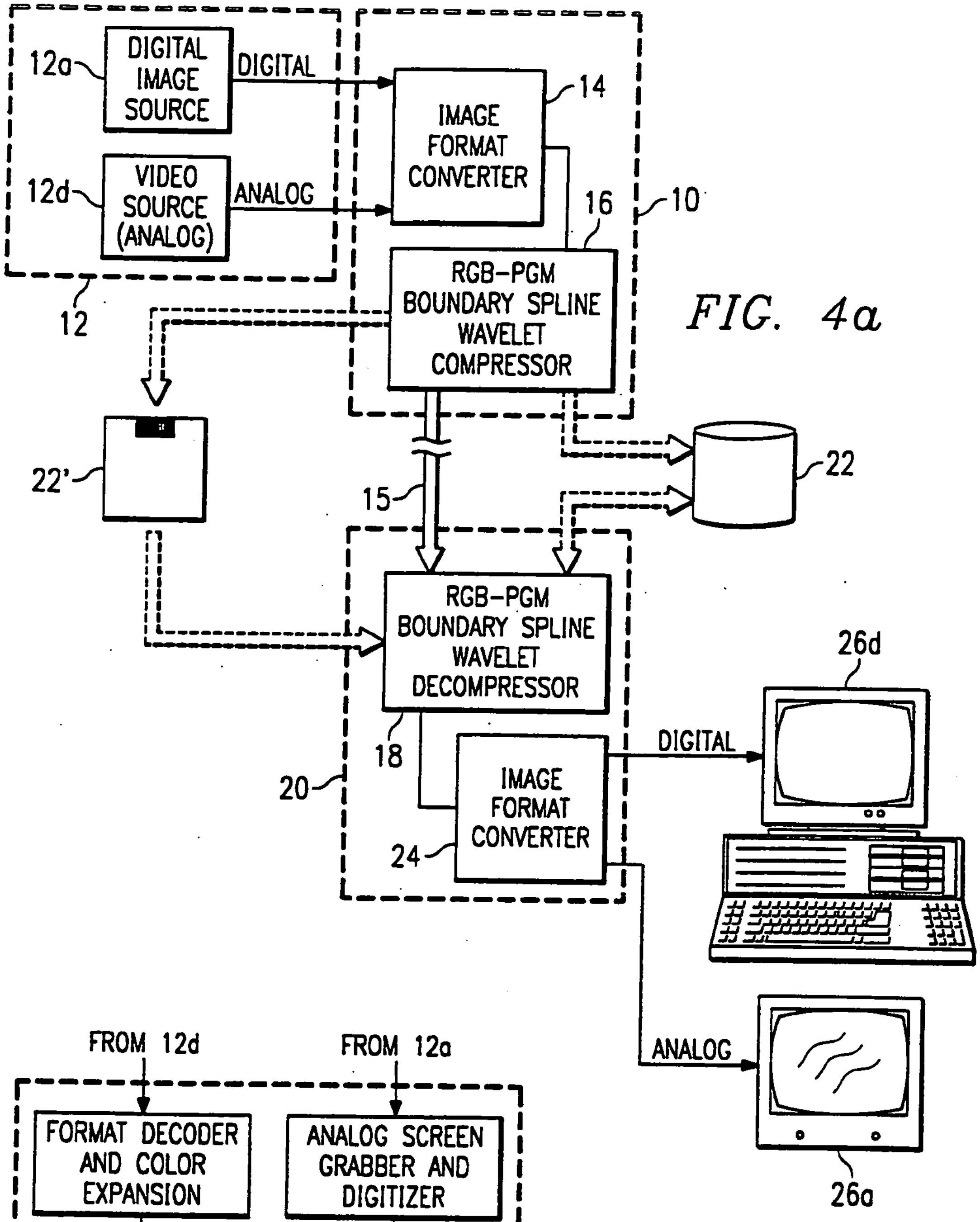

12a
DIGITAL IMAGE SOURCE
DIGITAL
12d
VIDEO SOURCE (ANALOG)
ANALOG
12
IMAGE FORMAT CONVERTER
14
16
10
RGB-PGM BOUNDARY SPLINE WAVELET COMPRESSOR
22'
15
22
RGB-PGM BOUNDARY SPLINE WAVELET DECOMPRESSOR
18
20
IMAGE FORMAT CONVERTER
24
DIGITAL
26d
ANALOG
26a

FIG. 4a

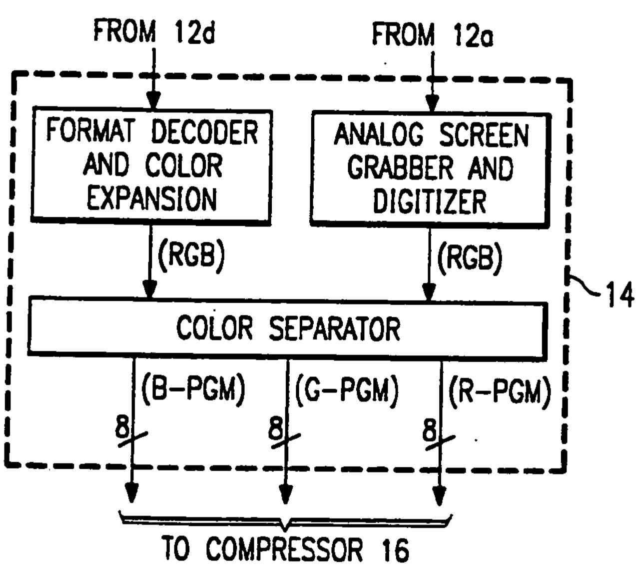

FROM 12d
FROM 12a
FORMAT DECODER AND COLOR EXPANSION
ANALOG SCREEN GRABBER AND DIGITIZER
(RGB)
(RGB)
14
COLOR SEPARATOR
(B-PGM)
(G-PGM)
(R-PGM)
8
8
8
TO COMPRESSOR 16

FIG. 4b

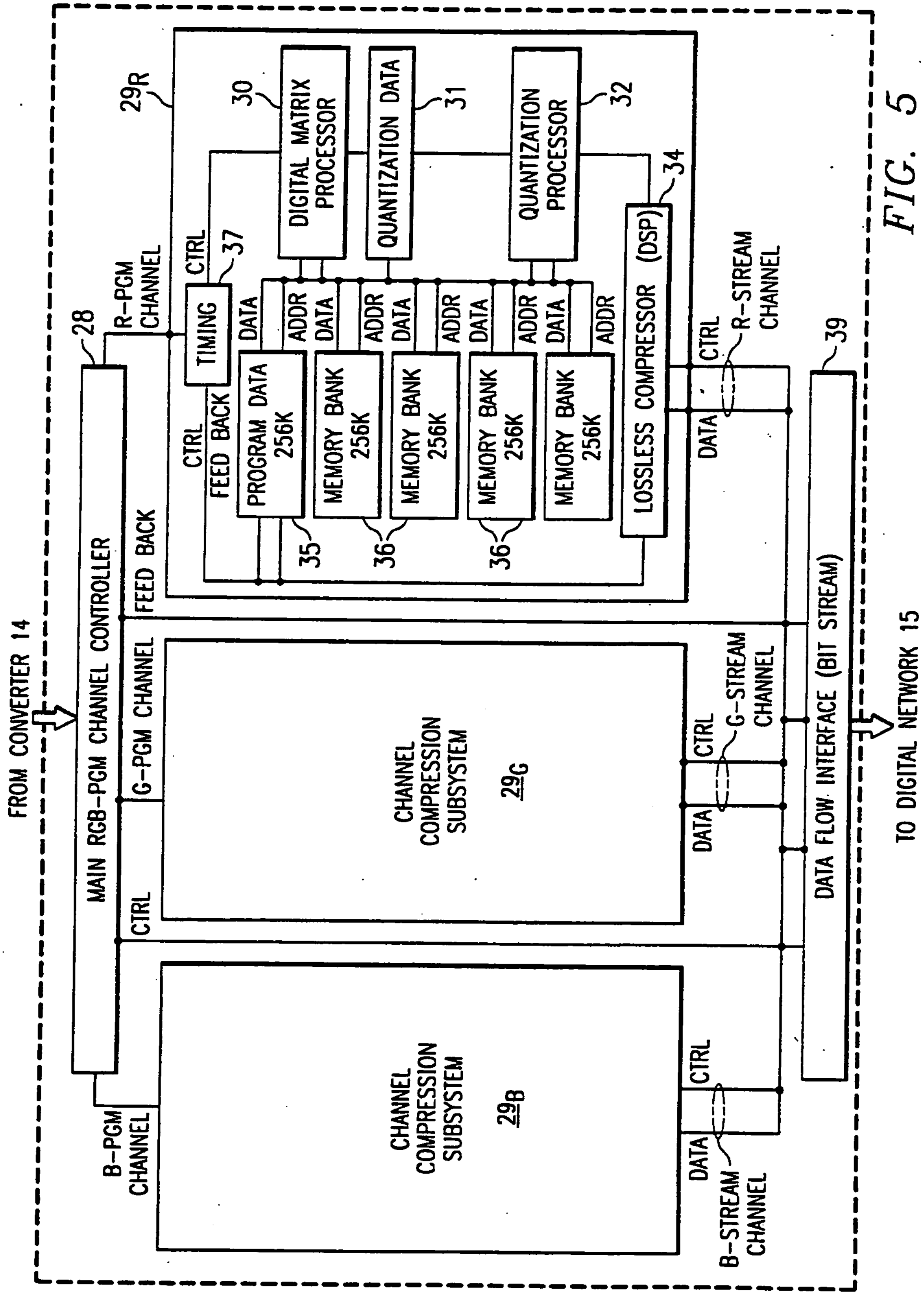

FROM CONVERTER 14
MAIN RGB-PGM CHANNEL CONTROLLER
28
R-PGM CHANNEL
G-PGM CHANNEL
B-PGM CHANNEL
CTRL
FEED BACK
29R
TIMING
37
DATA
ADDR
PROGRAM DATA 256K
MEMORY BANK 256K
35
36
DIGITAL MATRIX PROCESSOR
30
QUANTIZATION DATA
31
QUANTIZATION PROCESSOR
32
LOSSLESS COMPRESSOR (DSP)
34
R-STREAM CHANNEL
G-STREAM CHANNEL
B-STREAM CHANNEL
CHANNEL COMPRESSION SUBSYSTEM
29G
29B
DATA FLOW INTERFACE (BIT STREAM)
39
TO DIGITAL NETWORK 15

FIG. 5

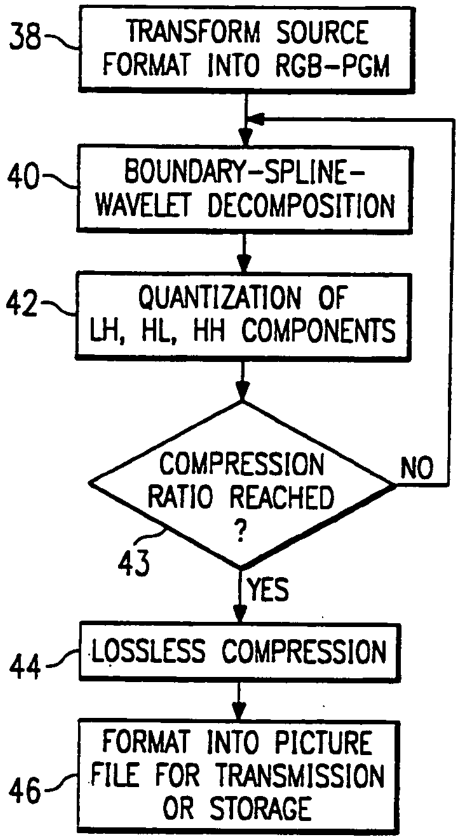


FIG. 6

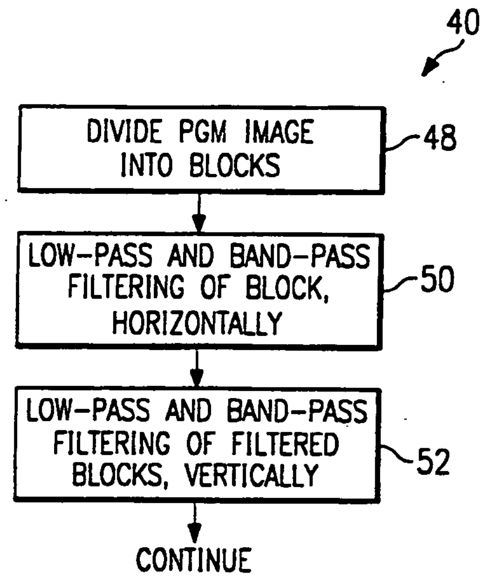


FIG. 7

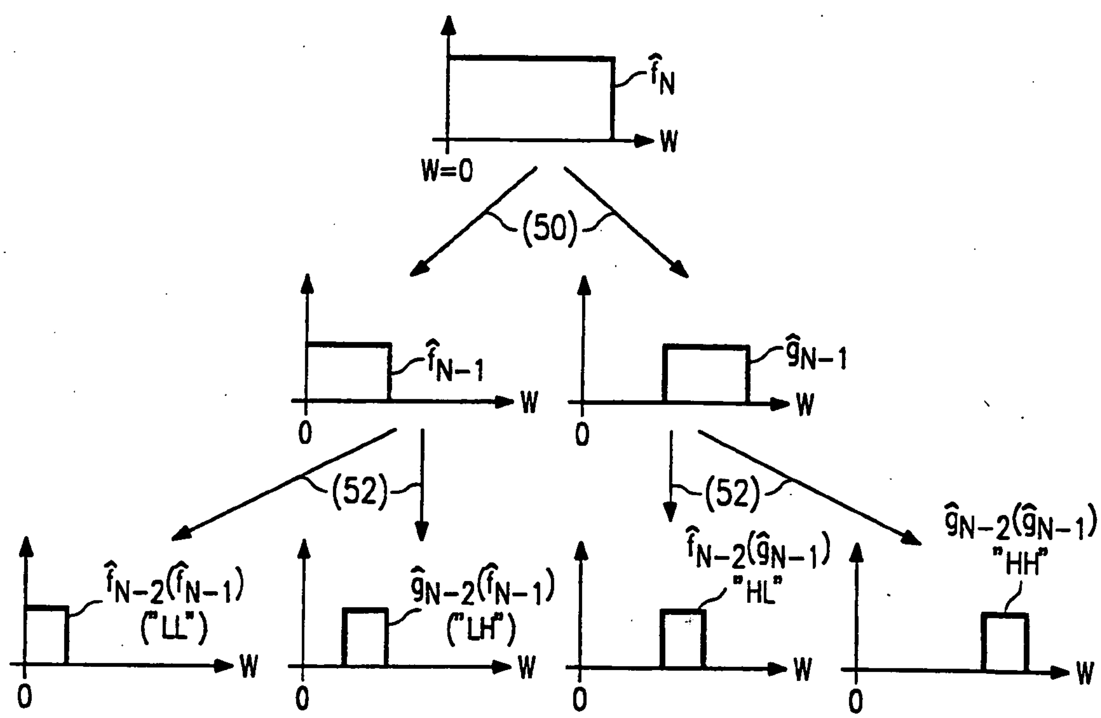


FIG. 8

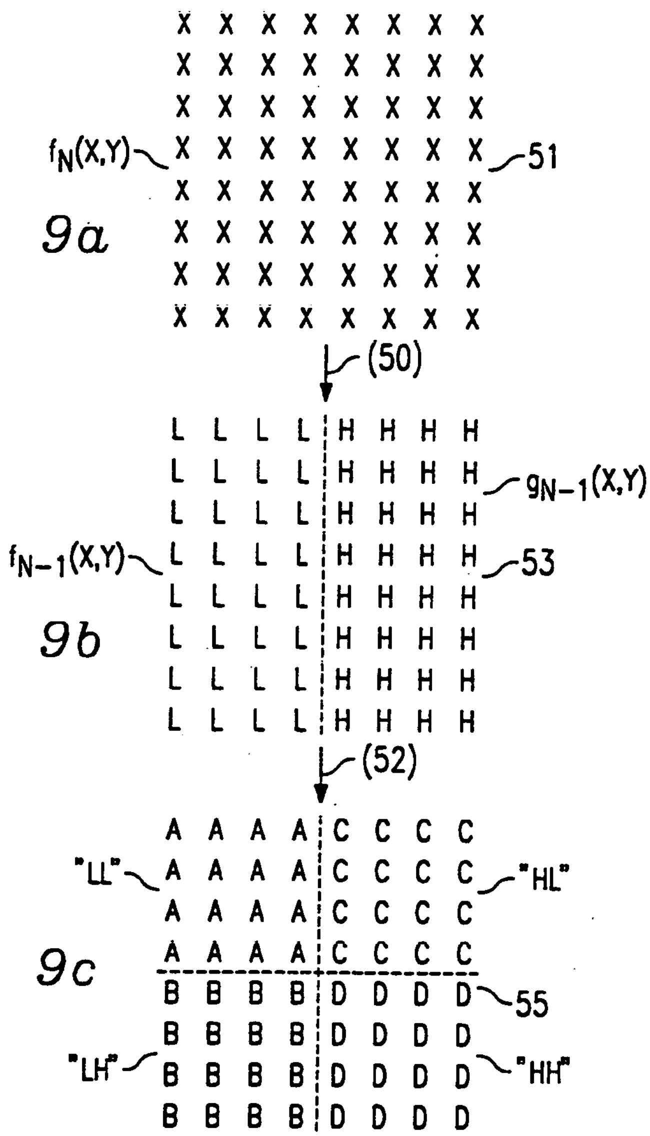
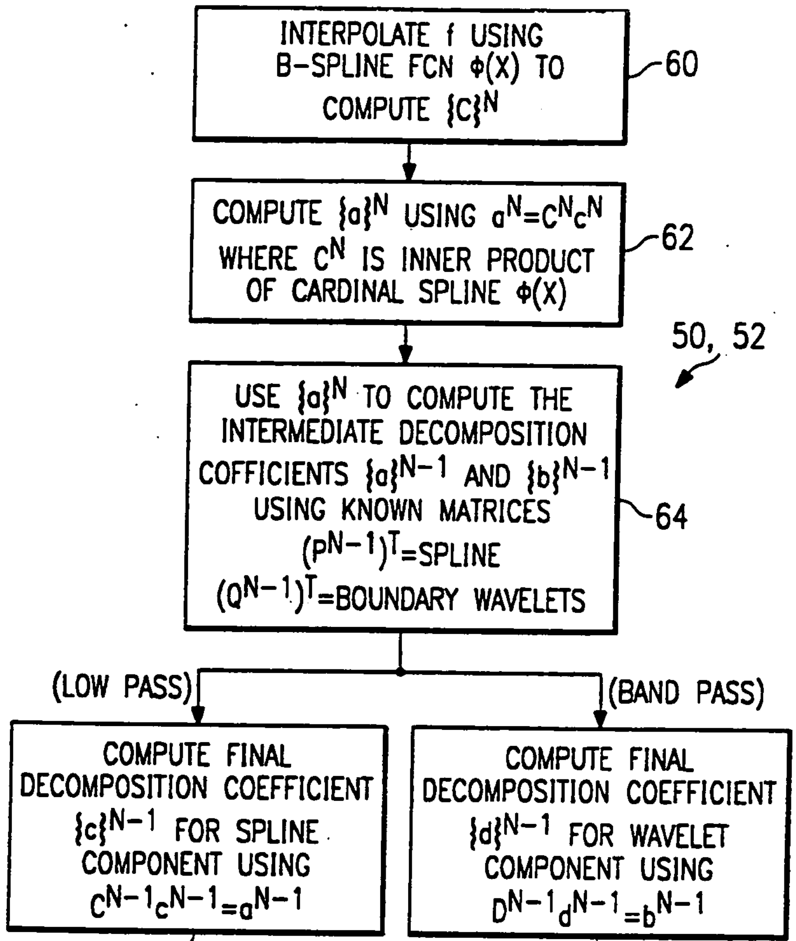


FIG. 9a

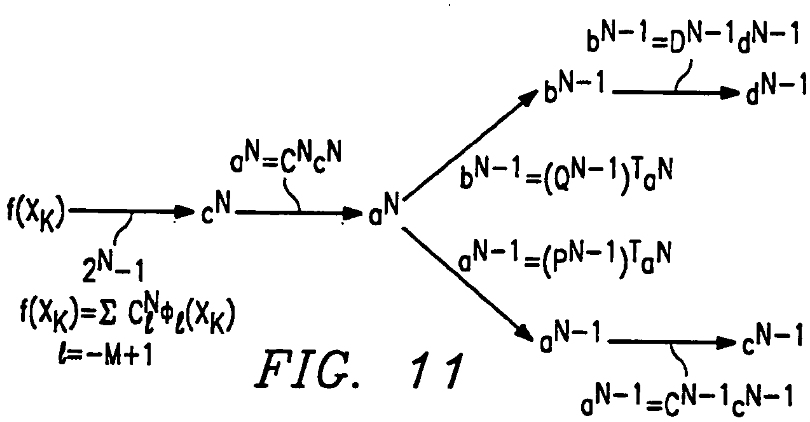
FIG. 9b

FIG. 9c



INTERPOLATE f USING B-SPLINE FCN Φ(X) TO COMPUTE {C}^N
60
COMPUTE {a}^N USING a^N=C^N c^N WHERE C^N IS INNER PRODUCT OF CARDINAL SPLINE Φ(X)
62
50, 52
USE {a}^N TO COMPUTE THE INTERMEDIATE DECOMPOSITION COFFICIENTS {a}^N−1 AND {b}^N−1 USING KNOWN MATRICES (P^N−1)^T=SPLINE (Q^N−1)^T=BOUNDARY WAVELETS
64
(LOW PASS)
(BAND PASS)
COMPUTE FINAL DECOMPOSITION COEFFICIENT {c}^N−1 FOR SPLINE COMPONENT USING C^N−1 c^N−1=a^N−1
COMPUTE FINAL DECOMPOSITION COEFFICIENT {d}^N−1 FOR WAVELET COMPONENT USING D^N−1 d^N−1=b^N−1
66
68

FIG. 10



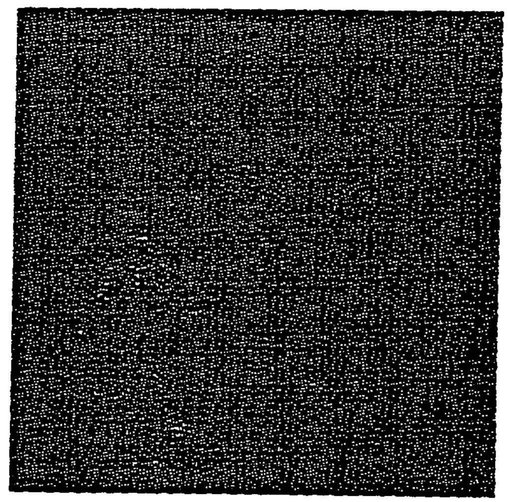
b^N−1=D^N−1 d^N−1
b^N−1
d^N−1
a^N=C^N c^N
b^N−1=(Q^N−1)^T a^N
f(X_K)
c^N
a^N
a^N−1=(P^N−1)^T a^N
2^N−1
f(X_K)=Σ C_l^N φ_l(X_K)
l=−M+1
a^N−1
c^N−1
a^N−1=C^N−1 c^N−1

FIG. 11

FIG. 12a

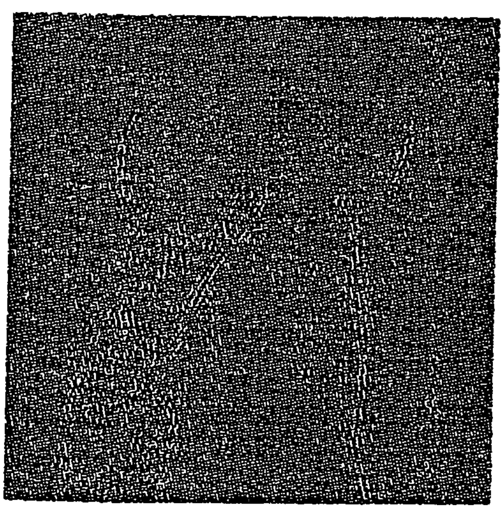
LL

FIG. 12b



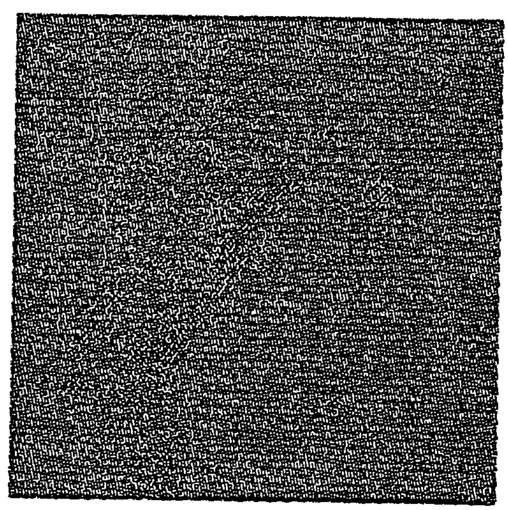
HL

FIG. 12d



LH

FIG. 12c

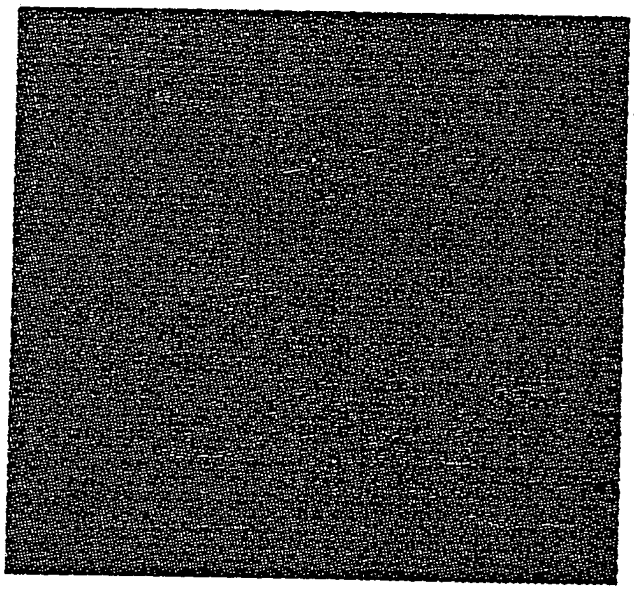


HH

FIG. 12e

LL

FIG. 13a



LH+HL+HH

FIG. 13b

LH+HL+HH AFTER THRESHOLDING

FIG. 13c

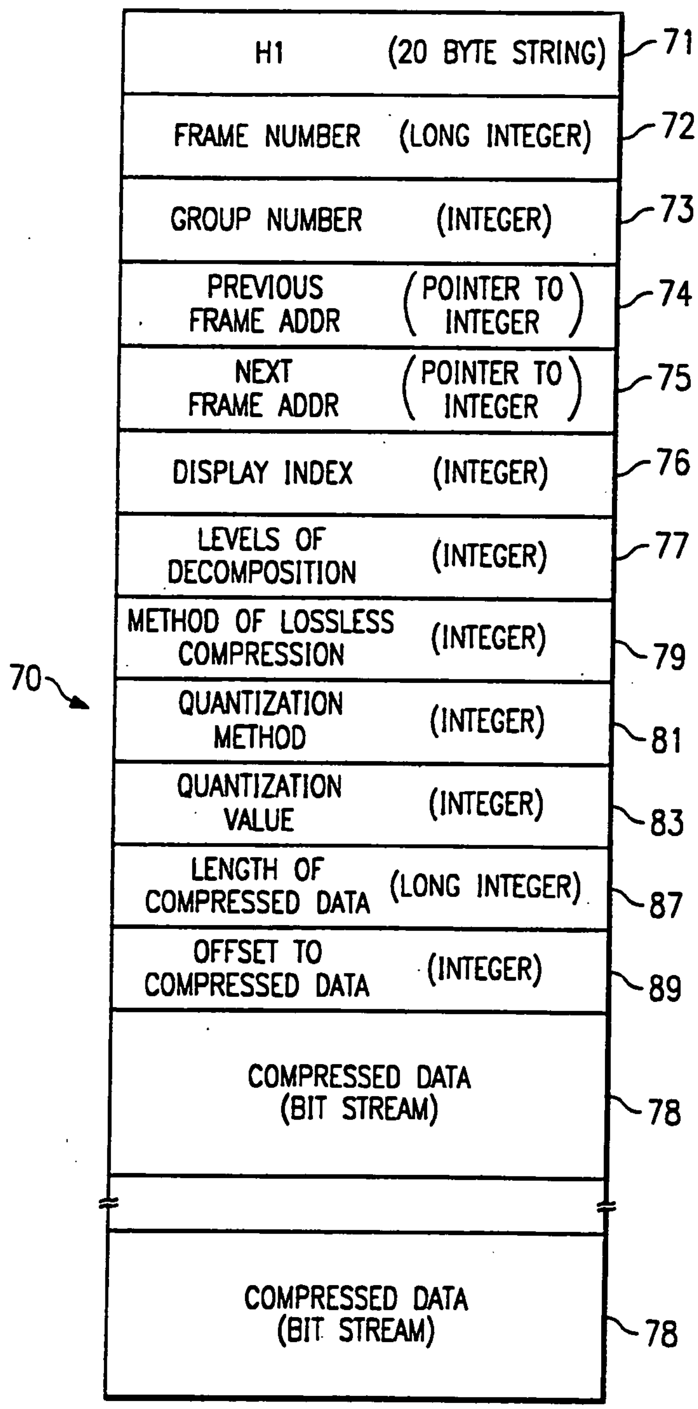

H1 (20 BYTE STRING)
71
FRAME NUMBER (LONG INTEGER)
72
GROUP NUMBER (INTEGER)
73
PREVIOUS FRAME ADDR (POINTER TO INTEGER)
74
NEXT FRAME ADDR (POINTER TO INTEGER)
75
DISPLAY INDEX (INTEGER)
76
LEVELS OF DECOMPOSITION (INTEGER)
77
METHOD OF LOSSLESS COMPRESSION (INTEGER)
79
70
QUANTIZATION METHOD (INTEGER)
81
QUANTIZATION VALUE (INTEGER)
83
LENGTH OF COMPRESSED DATA (LONG INTEGER)
87
OFFSET TO COMPRESSED DATA (INTEGER)
89
COMPRESSED DATA (BIT STREAM)
78
COMPRESSED DATA (BIT STREAM)
78

FIG. 14

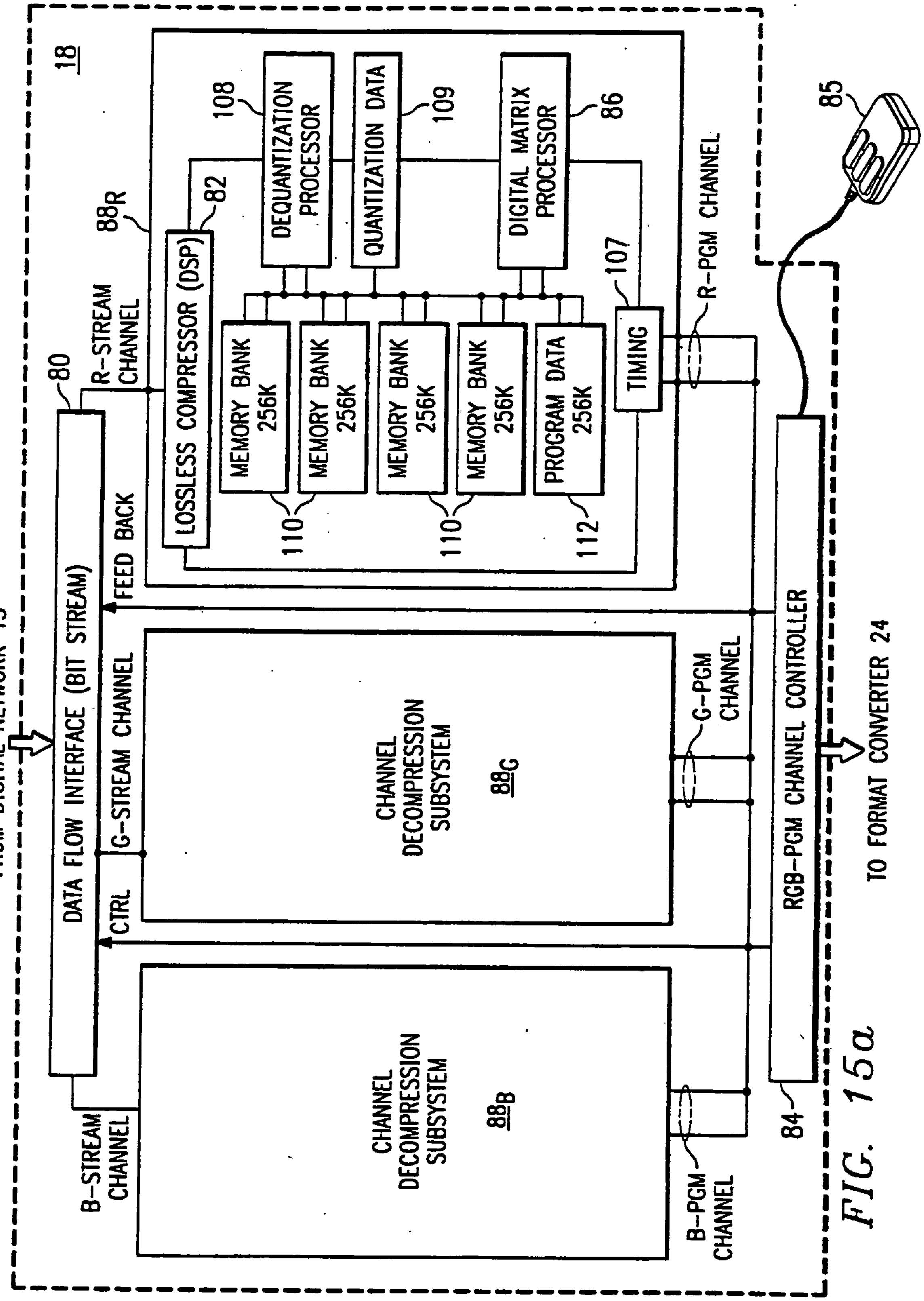

18
FROM DIGITAL NETWORK 15
DATA FLOW INTERFACE (BIT STREAM)
80
R-STREAM CHANNEL
88R
LOSSLESS COMPRESSOR (DSP)
82
DEQUANTIZATION PROCESSOR
108
QUANTIZATION DATA
109
DIGITAL MATRIX PROCESSOR
86
107
TIMING
MEMORY BANK 256K
MEMORY BANK 256K
MEMORY BANK 256K
MEMORY BANK 256K
PROGRAM DATA 256K
110
110
112
R-PGM CHANNEL
85
FEED BACK
G-STREAM CHANNEL
CHANNEL DECOMPRESSION SUBSYSTEM
88G
G-PGM CHANNEL
CTRL
CHANNEL DECOMPRESSION SUBSYSTEM
88B
B-STREAM CHANNEL
B-PGM CHANNEL
RGB-PGM CHANNEL CONTROLLER
84
TO FORMAT CONVERTER 24

FIG. 15a

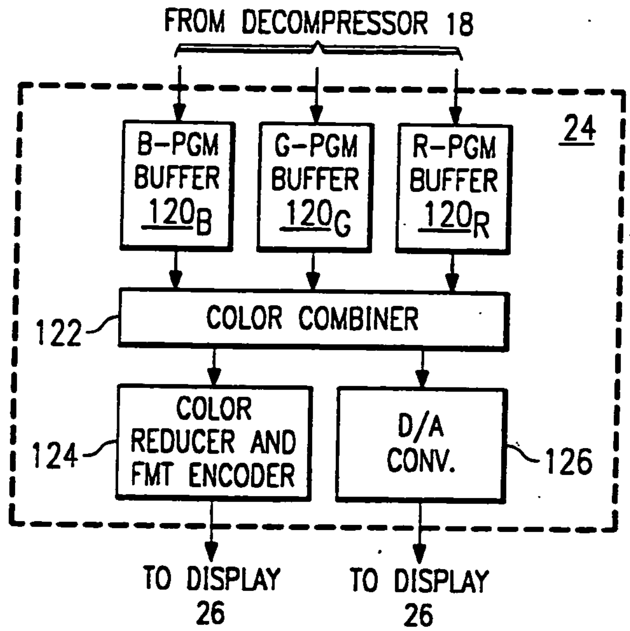


FIG. 15b

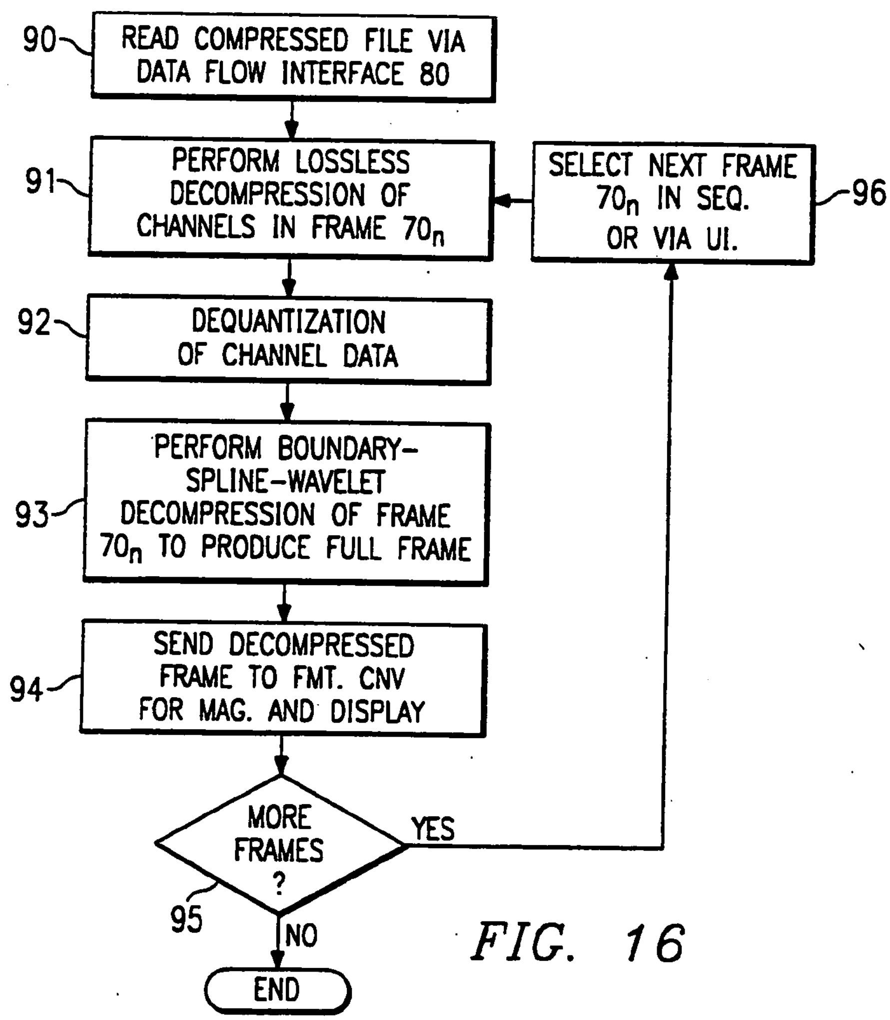


FIG. 16

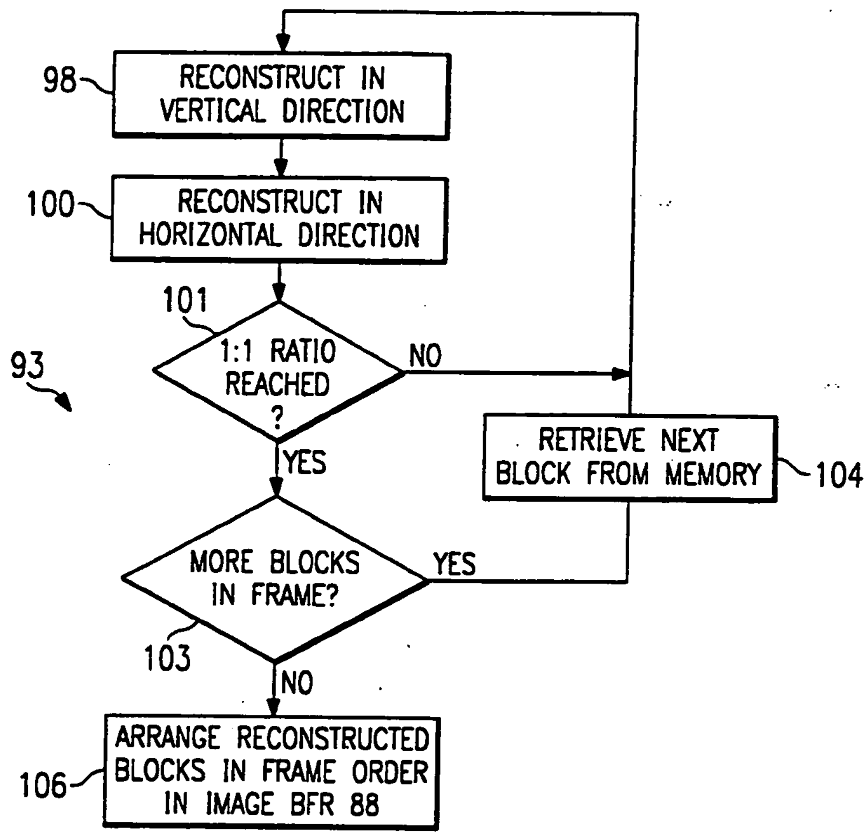

98
RECONSTRUCT IN VERTICAL DIRECTION
100
RECONSTRUCT IN HORIZONTAL DIRECTION
101
1:1 RATIO REACHED ?
NO
YES
93
RETRIEVE NEXT BLOCK FROM MEMORY
104
MORE BLOCKS IN FRAME?
YES
103
NO
ARRANGE RECONSTRUCTED BLOCKS IN FRAME ORDER IN IMAGE BFR 88
106

FIG. 17

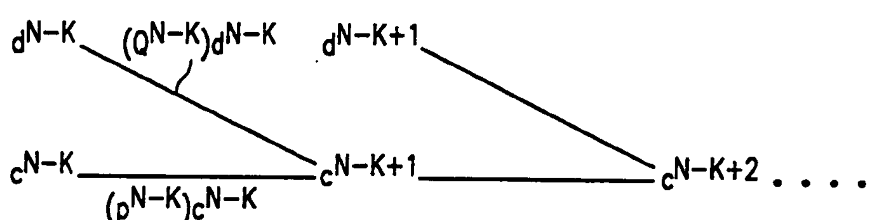
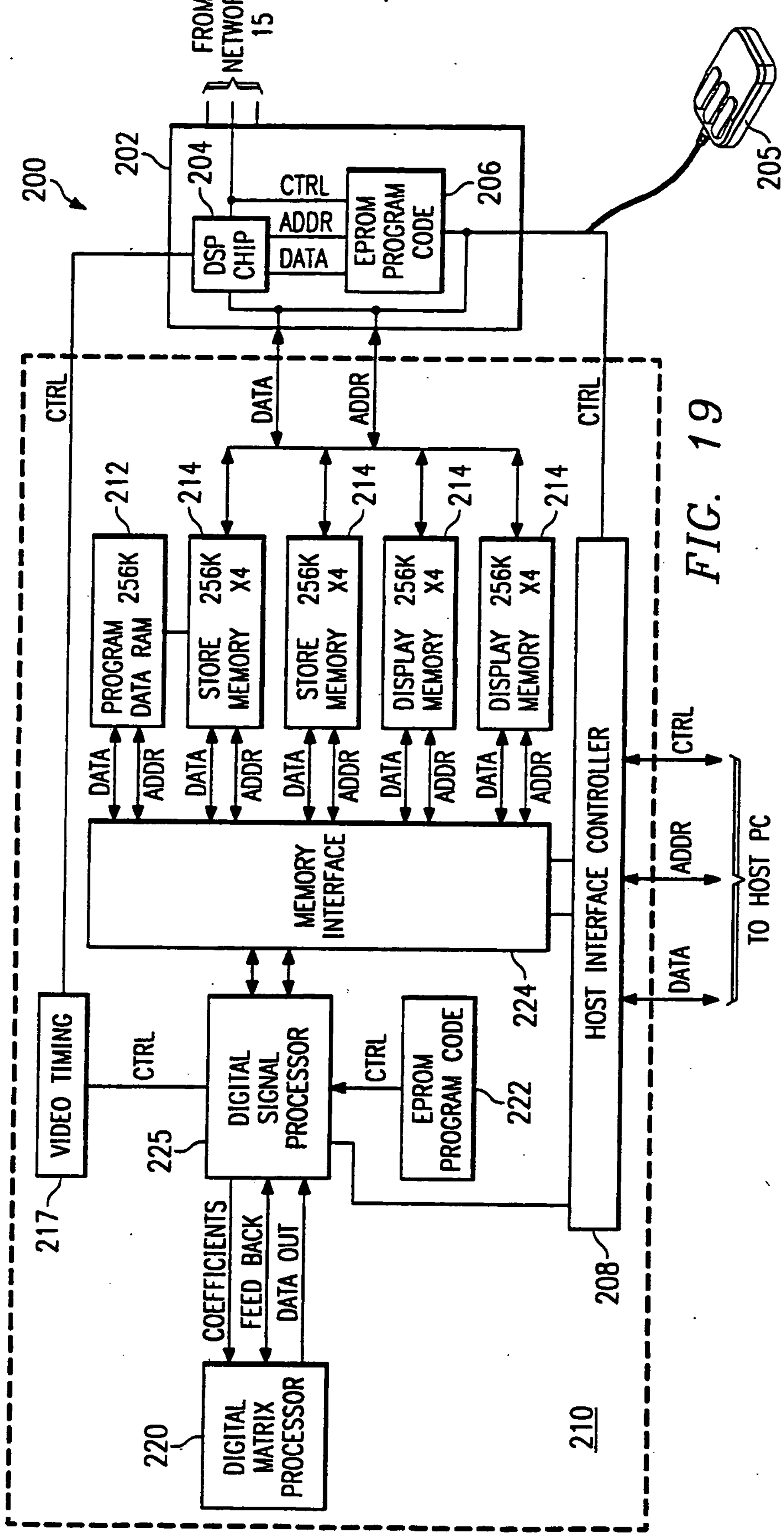

dN-K
(QN-K)dN-K
dN-K+1
cN-K
cN-K+1
cN-K+2 . . . .
(pN-K)cN-K

FIG. 18



FROM NETWORK 15
200
202
204
DSP CHIP
CTRL
ADDR
DATA
EPROM PROGRAM CODE
206
205
CTRL
DATA
ADDR
CTRL
PROGRAM DATA RAM 256K
212
STORE MEMORY 256K X4
214
STORE MEMORY 256K X4
214
DISPLAY MEMORY 256K X4
214
DISPLAY MEMORY 256K X4
214
DATA
ADDR
DATA
ADDR
DATA
ADDR
DATA
ADDR
DATA
ADDR
MEMORY INTERFACE
224
HOST INTERFACE CONTROLLER
CTRL
ADDR
DATA
TO HOST PC
VIDEO TIMING
217
CTRL
DIGITAL SIGNAL PROCESSOR
225
CTRL
EPROM PROGRAM CODE
222
208
COEFFICIENTS
FEED BACK
DATA OUT
DIGITAL MATRIX PROCESSOR
220
210

FIG. 19

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04N7/26 G06T1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US,A,5 262 958 (CHUI ET AL) 16 November 1993 see column 1, line 24 - column 4, line 48 see column 23, line 25 - column 24, line 9 see claims 1-3 see figures	1,2,5,7
A		3,4,8, 10, 12-15, 19,23, 30,31, 33,36, 38,39

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
19 May 1995	24.05.95
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Dippel, U

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SIGNAL PROCESSING VI - PROCEEDINGS OF EUSIPCO-92, vol. I, 24 August 1992 BRUXELLES, BE, pages 231-234, XP 000350395 AASE ET AL 'Image Subband Coding Using Short-Kernel Cosine, Modulated Filter Banks' see page 231, left column, paragraph 1 - paragraph 3 see page 232, right column, paragraph 3	1,13,19, 33
A	NUMERICAL METHODS OF APPROXIMATION THEORY, vol. 9, 1992 BASEL, CH, pages 53-75, CHUI ET AL 'Wavelets on a Bounded Interval' cited in the application see figures 2,3,6	1,5,8, 13-15, 19,23, 30,33,36
P,X	EP,A,0 622 741 (KLICS LTD) 2 November 1994	1-10, 12-14, 16-19, 22-39
	see page 7, line 12 - page 9, line 44 see figures	
P,A		11,15, 20,21
	& WO,A,94 23385 (KLICS LTD) 13 October 1994	
P,A	EP,A,0 611 051 (CANON KABUSHIKI KAISHA) 17 August 1994	

1

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
US-A-5262958	16-11-93	NONE		
EP-A-622741	02-11-94	AU-B-	6381394	24-10-94
		WO-A-	9423385	13-10-94
EP-A-611051	17-08-94	JP-A-	6223172	12-08-94

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☑ SKEWED/SLANTED IMAGES
- ☑ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

HH_{00}	GH_{00}	HH_{01}	GH_{01}	HH_{02}	GH_{02}	HH_{03}	GH_{03}
HG_{00}	GG_{00}	HG_{01}	GG_{01}	HG_{02}	GG_{02}	HG_{03}	GG_{03}
HH_{10}	GH_{10}	HH_{11}	GH_{11}	HH_{12}	GH_{12}	HH_{13}	GH_{13}
HG_{10}	GG_{10}	HG_{11}	GG_{11}	HG_{12}	GG_{12}	HG_{13}	GG_{13}
HH_{20}	GH_{20}	HH_{21}	GH_{21}	HH_{22}	GH_{22}	HH_{23}	GH_{23}
HG_{20}	GG_{20}	HG_{21}	GG_{21}	HG_{22}	GG_{22}	HG_{23}	GG_{23}
HH_{30}	GH_{30}	HH_{31}	GH_{31}	HH_{32}	GH_{32}	HH_{33}	GH_{33}
HG_{30}	GG_{30}	HG_{31}	GG_{31}	HG_{22}	GG_{32}	HG_{33}	GG_{33}

FIG. 16

OCTAVE O DECOMPOSITION AFTER FIRST CONV_ROW AND CONV_COL PASS

Time t	0	1	2	3	4	5
Input Data Value	HH_{00}	HH_{01}	HH_{02}	HH_{03}	HH_{10}	HH_{11}
muxsel(1)	1	1	1	1	1	1
muxsel(2)	2	1	1	3	2	1
muxsel(3)	3	2	2	4	3	2
andsel(1),(4)	pass	zero	pass	zero	pass	zero
andsel(2),(3)	zero	pass	pass	pass	zero	pass
addsel(1)	add	add	add	add	add	add
addsel(2)	add	sub	add	sub	add	sub
addsel(3)	add	add	add	add	add	add
addsel(4)	sub	add	sub	add	sub	add
center muxsel (1)	l	r	l	r	l	r
center muxsel (2)	r	l	r	l	r	l
muxandsel (1)	zero	pass	pass	pass	zero	pass
muxandsel (2)	pass	pass	pass	pass	pass	pass
muxandsel (3)	zero	pass	pass	pass	zero	pass
OUT 2				$32HHH_{00}$		$32HHH_{01}$
OUT 1				$32HHG_{00}$		$32HHG_{01}$
OUTPUT LEADS 520				HHH_{00}	HHG_{00}	HHH_{01}

FIG. 17A

CONV_ROW CONTROL SIGNALS AND OUTPUTS DURING THE FORWARD OCTAVE 1 TRANSFORM

6	7	8	9	10	11	
HH_{12}	HH_{13}	HH_{20}	HH_{21}	HH_{22}	HH_{23}	
1	1	1	1	1	1	
1	3	2	1	1	1	
2	4	3	2	2	2	
pass	zero	pass	zero	pass	zero	
pass	pass	zero	pass	pass	pass	
add	add	add	add	add	add	
add	sub	add	sub	add	sub	
add	add	add	add	add	add	
sub	add	sub	add	sub	add	
l	r	l	r	l	r	
r	l	r	l	r	l	
pass	pass	zero	pass	pass	pass	
pass	pass	pass	pass	pass	pass	
pass	pass	zero	pass	pass	pass	
	$32HHH_{10}$		$32HHH_{11}$		$32HHH_{20}$	
	$32HHG_{10}$		$32HHG_{11}$		$32HHG_{20}$	
HHG_{01}	HHH_{10}	HHG_{10}	HHH_{11}	HHG_{11}	HHH_{20}	HHG_{20}

KEY TO FIG. 17

FIG. 17A	FIG. 17B

FIG. 17

FIG. 17B

CONV_ROW CONTROL SIGNALS AND OUTPUTS DURING THE FORWARD OCTAVE 1 TRANSFORM

Time	Input Data Value	Output of Block 926	Output of Block 928
0	HH_{00}	0	$32\{(a+b)\ HH_{00}\}$
1	HH_{01}	$32aHH_{01}$	$32\{(c+d)HH_{00} - bHH_{01}\}$
2	HH_{02}	$32HHG_{00} = 32\{(c+d)HH_{00} - bHH_{01} + aHH_{02}\}$	$32\{aHH_{01} + bHH_{02}\}$
3	HH_{03}	0	$32\{dHH_{01} + cHH_{02} - (b-a)HH_{03}\}$
4	HH_{10}	$32HHG_{01} = 32\{dHH_{01} + cHH_{02} - (b-a)HH_{03}\}$	$32\{(a+b)HH_{10}\}$
5	HH_{11}	$32aHH_{11}$	$32\{(c+d)HH_{10} - bHH_{11}\}$
6	HH_{12}	$32HHG_{10} = 32\{(c+d)HH_{10} - bHH_{11} + aHH_{12}\}$	$32\{aHH_{11} + bHH_{12}\}$
	. . . . .	. . . . .	. . . . .

FIG. 18A

CONV_ROW DATA FLOW DURING THE FORWARD OCTAVE 1 TRANSFORM

Output of Block 930	Output of Block 932
$32(c+d)\,HH_{00}$	0
$32\{(a+b)HH_{00} + cHH_{01}\}$	$32dHH_{01}$
$32\{dHH_{01} + cHH_{02}\}$	$32\{(a+b)HH_{00} + cHH_{01} - dHH_{02}\}=32HHH_{00}$
$32\{aHH_{01} + bHH_{02} + (c-d)HH_{03}\}$	0
$32(c+d)HH_{10}$	$32\{aHH_{01} + bHH_{02} + (c-d)HH_{03}\} =32HHH_{01}$
$32\{(a+b)HH_{10}+cHH_{11}\}$	$32dHH_{11}$
$32\{dHH_{11} + cHH_{12}\}$	$32\{(a+b)HH_{10} + cHH_{11} - dHH_{12}\}=32HHH_{10}$
⋮	⋮

KEY TO FIG. 18

FIG. 18A	FIG. 18B

FIG. 18

FIG. 18B

CONV_ROW DATA FLOW DURING THE FORWARD OCTAVE 1 TRANSFORM

Time t	0 3	4 7	8 11
Input Data Values	HHH_{00} ··· HHG_{01}	HHH_{10} ·· HHG_{11}	HHH_{20} ·· HHG_{21}
muxsel(1)	1 1	1 1	1 1
muxsel(2)	2 2	1 1	1 1
muxsel(3)	3 3	2 2	2 2
andsel(1),(4)	pass .... pass	zero .... zero	pass .... pass
andsel(2),(3)	zero .... zero	pass .... pass	pass .... pass
addsel(1)	add .... add	add .... add	add .... add
addsel(2)	add .... add	sub .... sub	add .... add
addsel(3)	add .... add	add .... add	add .... add
addsel(4)	sub .... sub	add .... add	sub .... sub
centermuxel(1)	l l	r .... r	l l
centermuxel(2)	r .... r	l l	r .... r
muxandsel(1)	zero .... zero	pass .... pass	pass .... pass
muxandsel(2)	pass .... pass	pass .... pass	pass .... pass
muxandsel(3)	zero .... zero	pass .... pass	pass .... pass
OUT2			$32HHHH_{00}$ ·· $32HHGH_{01}$
OUT1			
OUTPUT LEADS 524			$HHHH_{00}$ ·· $HHGH_{01}$

FIG. 19A

CONV_COL CONTROL SIGNALS AND OUTPUTS DURING THE FORWARD OCTAVE 1 TRANSFORM

12 15	16 19	20 23
HHH_{30} .. HHG_{31}		
1 1	1 1	1 1
3 3	2 2	1 1
4 4	3 3	2 2
zero .... zero	pass .... pass	zero .... zero
pass .... pass	zero .... zero	pass .... pass
add .... add	add .... add	add .... add
sub .... sub	add .... add	sub .... sub
add .... add	add .... add	add .... add
add .... add	sub .... sub	add .... add
r .... r	l l	r .... r
l l	r .... r	l l
pass .... pass	zero .... zero	pass .... pass
pass .... pass	pass .... pass	pass .... pass
pass .... pass	zero .... zero	pass .... pass
$32HHHG_{00}$ .. $32HHGG_{01}$	$32HHHH_{10}$ .. $32HHGH_{11}$	$32HHHG_{10}$ .. $32HHGG_{11}$
$HHHG_{00}$ .. $HHGG_{01}$	$HHHH_{10}$ .. $HHGH_{11}$	$HHHG_{10}$ .. $HHGG_{11}$

FIG. 19B

KEY TO FIG 19

FIG. 19A	FIG. 19B

FIG 19

CONV_COL CONTROL SIGNALS AND OUTPUTS DURING THE FORWARD OCTAVE 1 TRANSFORM

t	Input Data Value	Output of Block 1326
0	HHH_{00}	. .
. .		. .
3	HHG_{01}	. .
4	HHH_{10}	$32aHHH_{10}$
. .		. .
7	HHG_{11}	$32aHHG_{11}$
8	HHH_{20}	$32HHHG_{00} = 32\{(c+d)HHH_{00} - bHHH_{10} + aHHH_{20}\}$
. .		. .
11	HHG_{21}	$32HHGG_{01} = 32\{(c+d)HHG_{01} - bHHG_{11} + aHHG_{21}\}$
12	HHH_{30}	. .
. .		. .
15	HHG_{31}	. .
16		$32HHHG_{10} = 32\{dHHH_{10} + cHHH_{20} - (b-a)HHH_{30}\}$
. .		. .
19		$32HHGG_{11} = 32\{dHHG_{11} + HHG_{21} - (b-a)HHG_{31}\}$

FIG. 20A

CONV_COL DATA FLOW FOR FORWARD OCTAVE 1 TRANSFORM

Output of Block 1328	Output of Block 1330
$32(a+b)HHH_{00}$	$32(c+d)HHH_{00}$
⋮	⋮
$32(a+b)HHG_{01}$	$32(c+d)HHG_{01}$
$32\{(c+d)HHH_{00} - bHHH_{10}\}$	$32\{(a+b)HHH_{00} + cHHH_{10}\}$
⋮	⋮
$32\{(c+d)HHG_{01} - bHHG_{11}\}$	$32\{(a+b)HHG_{01} + cHHG_{11}\}$
$32\{aHHH_{10} + bHHH_{20}\}$	$32\{dHHH_{10} + cHHH_{20}\}$
⋮	⋮
$32\{aHHG_{11} + bHHG_{21}\}$	$32\{dHHG_{11} + cHHG_{21}\}$
$32\{dHHH_{10} + cHHH_{20} - (b-a)HHH_{30}\}$	$32\{aHHH_{10} + bHHH_{20} + (c-d)HHH_{30}\}$
⋮	⋮
$32\{dHHG_{11} + cHHG_{21} - (b-a)HHG_{31}\}$	$32\{aHHG_{11} + bHHG_{20} + (c-d)HHG_{31}\}$

FIG. 20B

CONV_COL DATA FLOW FOR FORWARD OCTAVE 1 TRANSFORM

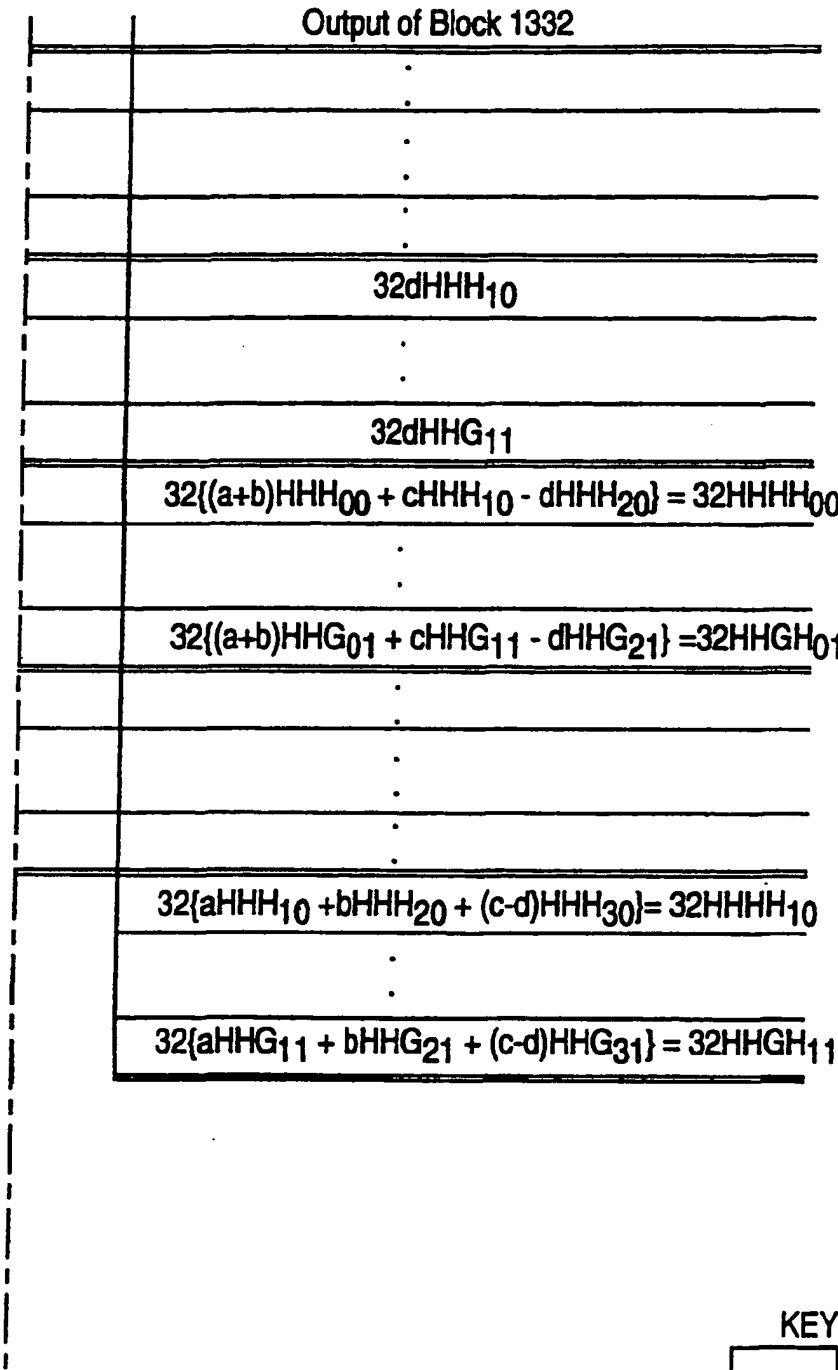


FIG. 20C

CONV_COL DATA FLOW FOR FORWARD OCTAVE 1 TRANSFORM

KEY TO FIG. 20

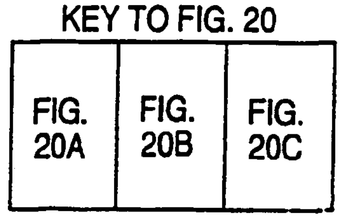


FIG. 20

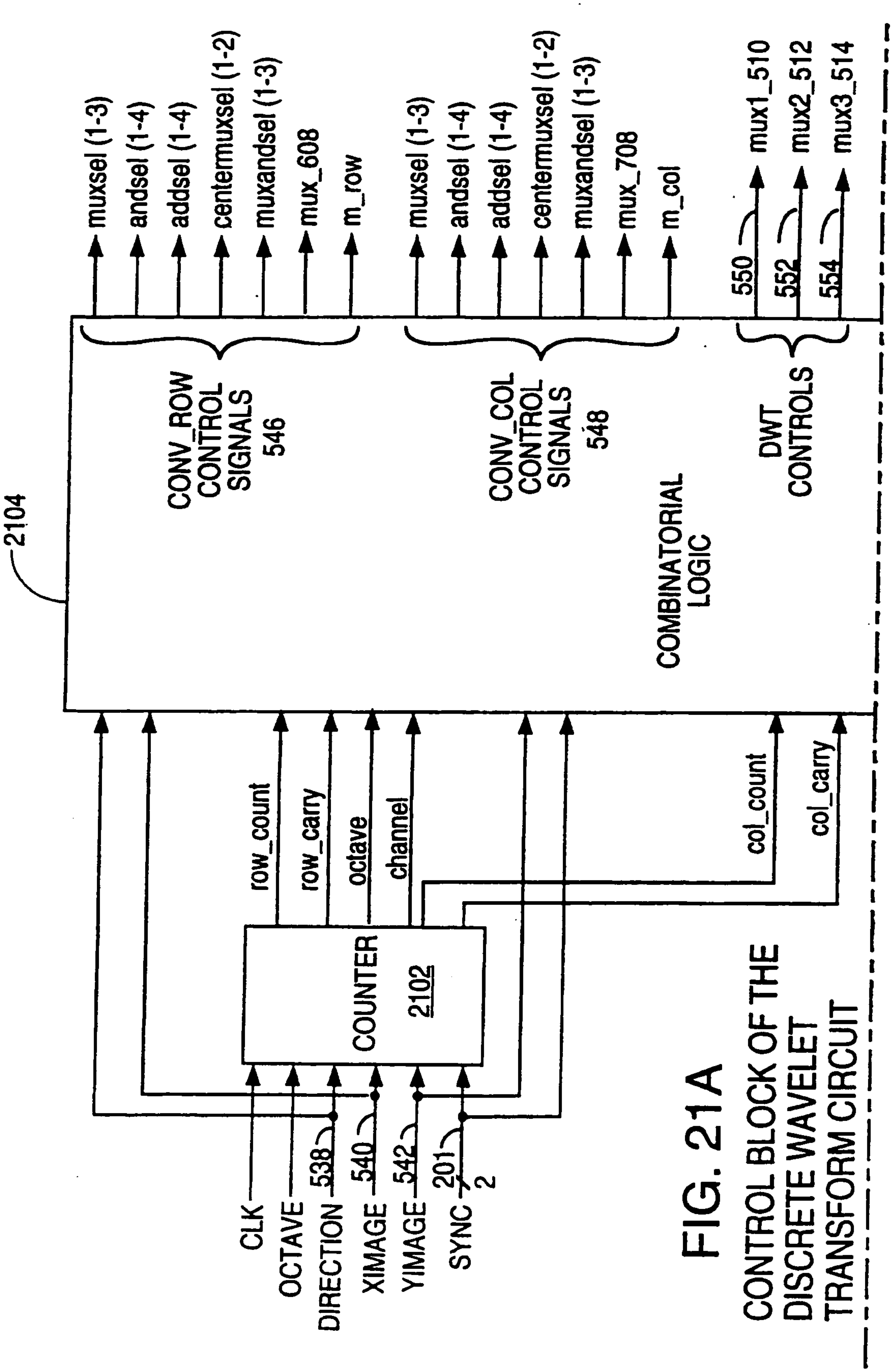


FIG. 21A

CONTROL BLOCK OF THE DISCRETE WAVELET TRANSFORM CIRCUIT

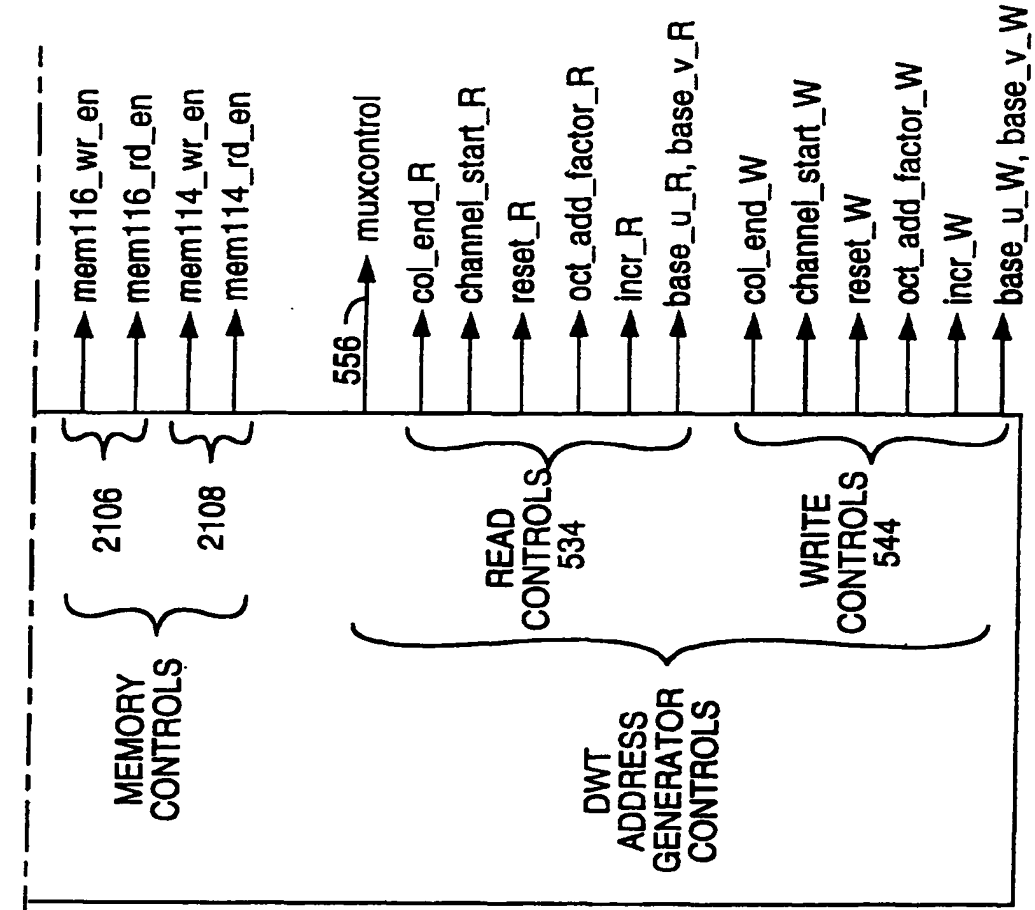


FIG. 21B

CONTROL BLOCK OF THE DISCRETE WAVELET TRANSFORM CIRCUIT

KEY TO FIG. 21

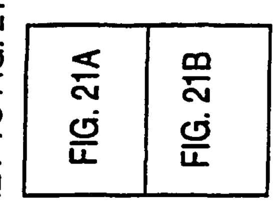


FIG. 21

Time t	0 3	4 7	8 11
muxsel(1)	2 2	4 4	3 3
muxsel(2)	3 3	1 1	1 1
muxsel(3)	1 1	1 1	1 1
andsel(1),(4)	zero .... zero	pass .... pass	zero .... zero
andsel(2)	zero .... zero	pass .... pass	pass .... pass
andsel(3)	pass .... pass	pass .... pass	pass .... pass
addsel(1)	add .... add	add .... add	add .... add
addsel(2)	add .... add	sub .... sub	add .... add
addsel(3)	add .... add	add .... add	add .... add
addsel(4)	sub .... sub	add .... add	sub .... sub
centermuxel(1)	r .... r	l l	r .... r
centermuxel(2)	l l	r .... r	l l
muxandsel(1)	pass .... pass	pass .... pass	pass .... pass
muxandsel(2)	zero .... zero	pass .... pass	pass .... pass
muxandsel(3)	pass .... pass	zero .... zero	pass .... pass
OUT2			
OUT1			$8HHH_{00}$ .. $8HHG_{01}$
OUTPUT LEADS 524			HHH_{00} .. HHG_{01}

FIG. 22A

CONV_COL CONTROL SIGNALS AND OUTPUTS FOR INVERSE OCTAVE 1 TRANSFORM

12 15	16 19	20 23
2 2	2 2	4 4
2 2	3 3	1 1
1 1	1 1	1 1
pass .... pass	zero .... zero	pass .... pass
pass .... pass	zero .... zero	pass .... pass
pass .... pass	pass .... pass	pass .... pass
add .... add	add .... add	add .... add
sub .... sub	add .... add	sub .... sub
add .... add	add .... add	add .... add
add .... add	sub .... sub	add .... add
l l	r .... r	l l
r .... r	l l	r .... r
pass .... pass	pass .... pass	pass .... pass
pass .... pass	zero .... zero	pass .... pass
pass .... pass	pass .... pass	zero .... zero
$16HHH_{10}$ .. $16HHG_{11}$	$16HHH_{20}$. $16HHG_{21}$	$8HHH_{30}$ .. $8HHG_{31}$
HHH_{10} ... HHG_{11}	HHH_{20} .... HHG_{21}	HHH_{30} .. HHG_{31}

FIG. 22B

CONV_COL CONTROL SIGNALS AND OUTPUTS FOR INVERSE OCTAVE 1 TRANSFORM

KEY TO FIG 22

FIG. 22A	FIG. 22B

FIG 22

Time t	Input Data Value	OUTPUT OF BLOCK 1326
0	$HHHH_{00}$	$32cHHHH_{00}$
.		.
.		.
3	$HHGH_{01}$	$32cHHGH_{01}$
4	$HHHG_{00}$	$8HHH_{00}=32\{(b-a)HHHH_{00}+(c-d)HHHG_{00}\}$
.		.
.		.
7	$HHGG_{01}$	$8HHG_{01}=32\{(b-a)HHGH_{01}+(c-d)HHGG_{01}\}$
8	$HHHH_{10}$	$32(c+d)HHHH_{10}$
.		.
.		.
11	$HHGH_{11}$	$32(c+d)HHGH_{11}$
12	$HHHG_{10}$	$16HHH_{20}=32\{-dHHHH_{00}+aHHHG_{00}+bHHHH_{10}+cHHGG_{10}\}$
.		.
.		.
15	$HHGG_{11}$	$16HHG_{21}=32\{-dHHGH_{01}+aHHGG_{01}+bHHGH_{11}+cHHHG_{11}\}$
16		
.		
.		
19		
20		
.		
.		
23		

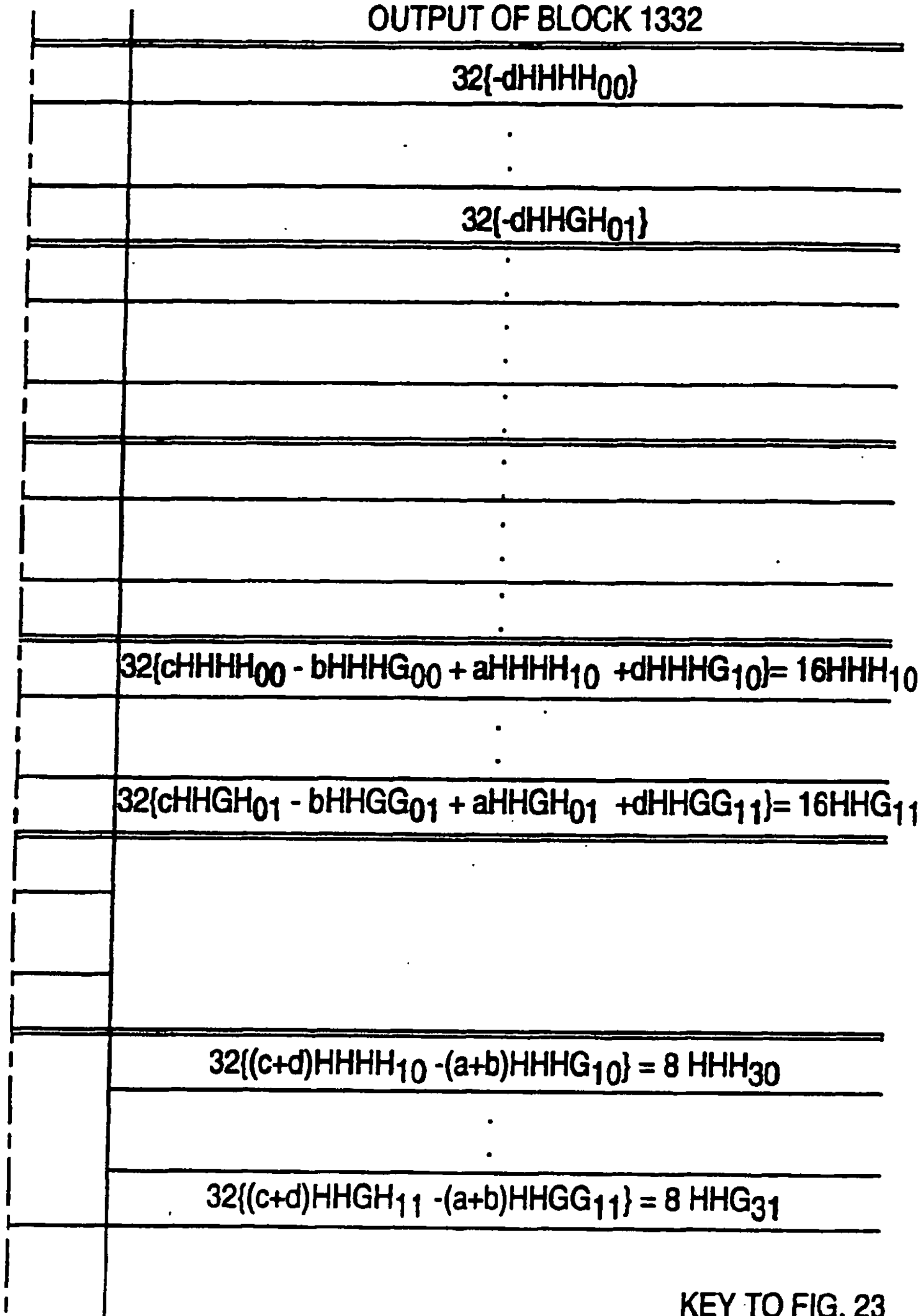
FIG. 23A

CONV_COL DATA FLOW FOR INVERSE OCTAVE 1 TRANSFORM

OUTPUT OF BLOCK 1328	OUTPUT OF BLOCK 1330
$32(b\text{-}a)HHHH_{00}$	$\vdots$
$\vdots$	$\vdots$
$32(b\text{-}a)HHGH_{01}$	$\vdots$
$32\{cHHHH_{00} - bHHHG_{00}\}$	$32\{-dHHHH_{00} + aHHHG_{00}\}$
$\vdots$	$\vdots$
$32\{cHHGH_{01} - bHHGG_{01}\}$	$32\{-dHHGH_{01} + aHHGG_{01}\}$
$32\{-dHHHH_{00} + aHHHG_{00} + bHHHH_{10}\}$	$32\{cHHHH_{00} - bHHHG_{00} + aHHHH_{10}\}$
$\vdots$	$\vdots$
$32\{-dHHGH_{01} + aHHGG_{01} + bHHGH_{11}\}$	$32\{cHHGH_{01} - bHHGG_{01} + aHHGH_{11}\}$
$32\{(c+d)HHHH_{10} - (a+b)HHHG_{10}\}$	$\vdots$
$\vdots$	$\vdots$
$32\{(c+d)HHGH_{11} - (a+b)HHGG_{11}\}$	$\vdots$
	$32\{(c+d)HHHH_{10} - (a+b)HHHG_{10}\}$
	$\vdots$
	$32\{(c+d)HHGH_{11} - (a+b)HHGG_{11}\}$

FIG. 23B

CONV_COL DATA FLOW FOR INVERSE OCTAVE 1 TRANSFORM



KEY TO FIG. 23

FIG. 23A	FIG. 23B	FIG. 23C

FIG. 23C

CONV_COL DATA FLOW FOR INVERSE OCTAVE 1 TRANSFORM

Time t	0	1	2	3	4	
INPUT DATA VALUE	HHH_{00}	HHG_{00}	HHH_{01}	HHG_{01}	HHH_{10}	
muxsel(1)	2	4	3	2	2	
muxsel(2)	3	1	1	2	3	
muxsel(3)	1	1	1	1	1	
andsel(1),(4)	zero	pass	zero	pass	zero	
andsel(2)	zero	pass	pass	pass	zero	
andsel(3)	pass	pass	pass	pass	pass	
addsel(1)	add	add	add	add	add	
addsel(2)	add	sub	add	sub	add	
addsel(3)	add	add	add	add	add	
addsel(4)	sub	add	sub	add	sub	
centermuxsel (1)	r	l	r	l	r	
centermuxsel (2)	l	r	l	r	l	
muxandsel (1)	pass	pass	pass	pass	pass	
muxandsel (2)	zero	pass	pass	pass	zero	
muxandsel (3)	pass	zero	pass	pass	pass	
OUT 2					$16HH_{01}$	
OUT 1			$8HH_{00}$		$16HH_{02}$	
OUTPUT LEADS 520				HH_{00}	HH_{01}	HH_{02}

FIG. 24

CONV_ROW CONTROL SIGNALS AND OUTPUTS FOR INVERSE OCTAVE 1 TRANSFORM

Time	Input Data Value	Output of Block 926	Output of Block 928
0	HHH_{00}	$32cHHH_{00}$	$32(b-a)HHH_{00}$
1	HHG_{00}	$8HH_{00}= 32\{(b-a)HHH_{00} +(c-d)HHG_{00}\}$	$32\{cHHH_{00} - bHHG_{00}\}$
2	HHH_{01}	$32(c+d)HHH_{01}$	$32\{-dHHH_{00} + aHHG_{00} + bHHH_{01}\}$
3	HHG_{01}	$16HH_{02} = 32\{-dHHH_{00} + aHHG_{00} + bHHH_{01} +cHHG_{01}\}$	$32\{(c+d)HHH_{03} - (a+b)HHG_{01}\}$
4	HHH_{10}	$32cHHH_{10}$	$32(b-a)HHH_{10}$
5	HHG_{10}	$8HH_{10} = 32\{(b-a)HHH_{10} +(c-d)HHG_{10}\}$	$32\{cHHH_{10} - bHHG_{10}\}$
6			

FIG. 25A

CONV_ROW DATA FLOW FOR INVERSE OCTAVE 1 TRANSFORM

Output of Block 930	Output of Block 932
	$-32dHHH_{00}$
$32\{-dHHH_{00} + aHHG_{00}\}$	
$32\{cHHH_{00} - bHHG_{00} + aHHH_{01}\}$	
	$16HH_{01} = 32\{cHHH_{00} - bHHG_{00} + aHHH_{01} + dHHG_{01}\}$
$32\{(c+d)HHH_{01} - (a+b)HHG_{01}\}$	$32\{-dHHH_{10}\}$
$32\{-dHHH_{10} + aHHG_{10}\}$	$8HH_{03} = 32\{(c+d)HHH_{01} - (a+b)HHG_{01}\}$

FIG. 25B

CONV_ROW DATA FLOW FOR INVERSE OCTAVE 1 TRANSFORM

KEY TO FIG. 25

FIG. 25A	FIG. 25B

FIG. 25

Time t	0 7	8 15	16 23
muxsel(1)	2 2	4 4	2 2
muxsel(2)	3 3	1 1	1 1
muxsel(3)	1 1	1 1	1 1
andsel(1),(4)	zero .... zero	pass .... pass	zero .... zero
andsel(2)	zero .... zero	pass .... pass	pass .... pass
andsel(3)	pass .... pass	pass .... pass	pass .... pass
addsel(1)	add .... add	add .... add	add .... add
addsel(2)	add .... add	sub .... sub	add .... add
addsel(3)	add .... add	add .... add	add .... add
addsel(4)	sub .... sub	add .... add	sub .... sub
centermuxel(1)	r .... r	l l	r .... r
centermuxel(2)	l l	r .... r	l l
muxandsel(1)	pass .... pass	pass .... pass	pass .... pass
muxandsel(2)	zero .... zero	pass .... pass	pass .... pass
muxandsel(3)	pass .... pass	zero .... zero	pass .... pass
out2			
out1			8H00 .... 8G03
OUTPUT LEADS 524			H00 .... G03

FIG. 26A

CONV_COL CONTROL SIGNALS AND OUTPUTS DURING INVERSE OCTAVE 0 TRANSFORM

24 31	32 39	....	48 55
2 2	2 2	....	3 3
1 1	1 1	....	1 1
1 1	1 1	....	1 1
pass .... pass	zero .... zero	....	zero .... zero
pass .... pass	pass .... pass	....	pass .... pass
pass .... pass	pass .... pass	....	pass .... pass
add .... add	add .... add	....	add .... add
sub .... sub	add .... add	....	add .... add
add .... add	add .... add	....	add .... add
add .... add	sub .... sub	....	sub .... sub
l l	r l	....	r .... r
r .... r	r l	....	l l
pass .... pass	pass .... pass	....	pass .... pass
pass .... pass	pass .... pass	....	pass .... pass
pass .... pass	pass .... pass	....	pass .... pass
$16H_{10}$ .. $16G_{13}$	$16H_{20}$ .... $16G_{23}$		$16H_{40}$ .... $16G_{43}$
H_{10} .... G_{13}	H_{20} .... G_{23}		H_{40} .... G_{43}

FIG. 26B

CONV_COL CONTROL SIGNALS AND OUTPUTS DURING INVERSE OCTAVE 0 TRANSFORM

56 63	64 71	72 79
2 2	2 2	4 4
2 2	3 3	1 1
1 1	1 1	1 1
pass .... pass	zero .... zero	pass .... pass
pass .... pass	zero .... zero	pass .... pass
pass .... pass	pass .... pass	pass .... pass
add .... add	add .... add	add .... add
sub .... sub	add .... add	sub .... sub
add .... add	add .... add	add .... add
add .... add	sub .... sub	add .... add
l l	r .... r	l l
r .... r	l l	r .... r
pass .... pass	pass .... pass	pass .... pass
pass .... pass	zero .... zero	pass .... pass
pass .... pass	pass .... pass	zero .... zero
$16H_{50}$ .... $16G_{53}$	$16H_{60}$ .... $16G_{63}$	$8H_{70}$ .... $8G_{73}$
H_{50} .... G_{53}	H_{60} .... G_{63}	H_{70} .... G_{73}

KEY TO FIG 26

FIG. 26A	FIG. 26B	FIG. 26C

FIG 26

FIG. 26C

CONV_COL CONTROL SIGNALS AND OUTPUTS DURING INVERSE OCTAVE 0 TRANSFORM

Time t	Input Data Value	Output of Block 1326
0	HH_{00}	$32cHH_{00}$
.		.
.		.
7	GH_{03}	$32cGH_{03}$
8	HG_{00}	$8H_{00} = 32\{(b-a)HH_{00} + (c-d)HG_{00}\}$
.		.
.		.
15	GG_{03}	$8G_{03} = 32\{(b-a)GH_{03} + (c-d)GG_{03}\}$
16	HH_{10}	$32cHH_{10}$
.		.
.		.
23	GH_{13}	$32cGH_{13}$
24	HG_{10}	$16H_{20} = 32\{-dHH_{00} + aHG_{00} + bHH_{10} + cHG_{10}\}$
.		.
.		.
31	GG_{13}	$16G_{23} = 32\{-dGH_{03} + aGG_{03} + bGH_{13} + cGG_{13}\}$
32	HH_{20}	$32cHH_{20}$
.		.
.		.
39	GH_{23}	$32cGH_{23}$

FIG. 27A

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

Output of Block 1328	Output of Block 1330
$32(b\text{-}a)HH_{00}$	. .
. .	. . .
$32(b\text{-}a)GH_{03}$	. .
$32cHH_{00} - 32bHG_{00}$	$-32dHH_{00} + 32aHG_{00}$
. .	. .
$32cGH_{03} - 32bGG_{03}$	$-32dGH_{03} + 32aGG_{03}$
$32\{-dHH_{00} + aHG_{00} + bHH_{10}\}$	$32\{cHH_{00} - bHG_{00} + aHH_{10}\}$
. .	. .
$32\{-dGH_{03} + aGG_{03} + bGH_{13}\}$	$32\{cGH_{03} - bGG_{03} + aGH_{13}\}$
$32cHH_{10} - 32bHG_{10}$	$-32dHH_{10} + 32aHG_{10}$
. .	. .
$32cGH_{13} - 32bGG_{13}$	$-32dGH_{13} + 32aGG_{13}$
$32\{-dHH_{10} + aHG_{10} + bHH_{20}\}$	$32\{cHH_{10} - bHG_{10} + aHH_{20}\}$
$32\{-dGH_{13} + aGG_{13} + bGH_{23}\}$	$32\{cGH_{13} - bGG_{13} + aGH_{23}\}$

FIG. 27B

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

	Output of Block 1330
	$-32dHH_{00}$
	. .
	$-32dGH_{03}$
	. .
	. .
	. .
	$-32dHH_{10}$
	. .
	$-32dGH_{13}$
	$32\{cHH_{00} - bHG_{00} + aHH_{10} + dHG_{10}\} = 16H_{10}$
	. .
	$32\{cGH_{03} - bGG_{03} + aGH_{13} + dGG_{13}\} = 16G_{13}$
	$-32dHH_{20}$
	. .
	$-32dGH_{23}$

FIG. 27C

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

48	HH_{30}	$32(c+d)HH_{30}$	
.		.	
.		.	
55	GH_{33}	$32(c+d)GH_{30}$	
56	HG_{30}	$16H_{60} = 32\{-dHH_{20} + aHG_{20} + bHH_{30} + cHG_{30}\}$	
.		.	
.		.	
63	GG_{33}	$16G_{63} = 32\{-dGH_{23} + aGG_{23} + bGH_{33} + cGG_{33}\}$	
64		.	
.		.	
.		.	
71		.	
72		.	
.		.	
.		.	
79		.	

FIG. 27D

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

$32\{-dHH_{20} + aHG_{20} + bHH_{30}\}$	$32\{cHH_{20} - bHG_{20} + aHH_{30}\}$
$\vdots$	$\vdots$
$32\{-dGH_{21} + aGG_{21} + bGH_{31}\}$	$32\{cGH_{21} - bGG_{21} + aGH_{31}\}$
$32(c+d)HH_{30} - 32(a+b)HG_{30}$	$\vdots$
$\vdots$	$\vdots$
$32(c+d)GH_{33} - 32(a+b)GG_{33}$	$\vdots$
$\vdots$	$32(c+d)HH_{30} - 32(a+b)HG_{30}$
$\vdots$	$\vdots$
$\vdots$	$32(c+d)GH_{33} - 32(a+b)GG_{33}$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
$\vdots$	$\vdots$

FIG. 27E

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

$32\{cHH_{20} - bHG_{20} + aHH_{30} + dHG_{30}\} = 16H_{50}$

$32\{cGH_{23} - bGG_{23} + aGH_{33} + dGG_{33}\} = 16G_{53}$

$32\{(c+d)HH_{30} - (a+b)HG_{30}\} = 8H_{70}$

$32\{(c+d)GH_{33} - (a+b)GG_{33}\} = 8G_{73}$

KEY TO FIG. 27

FIG. 27A	FIG. 27B	FIG. 27C
FIG. 27D	FIG. 27E	FIG. 27F

FIG. 27

FIG. 27F

CONV_COL DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

Time t	0	1	2	3	4	5
Input Data Value	H_{00}	G_{00}	H_{01}	G_{01}	H_{02}	G_{02}
muxsel(1)	2	4	2	2	2	2
muxsel(2)	3	1	1	1	1	1
muxsel(3)	1	1	1	1	1	1
andsel(1),(4)	zero	pass	zero	pass	zero	pass
andsel(2)	zero	pass	pass	pass	pass	pass
andsel(3)	pass	pass	pass	pass	pass	pass
addsel(1)	add	add	add	add	add	add
addsel(2)	add	sub	add	sub	add	sub
addsel(3)	add	add	add	add	add	add
addsel(4)	sub	add	sub	add	sub	add
centermuxsel (1)	r	l	r	l	r	l
centermuxsel (2)	l	r	l	r	l	r
muxandsel (1)	pass	pass	pass	pass	pass	pass
muxandsel (2)	zero	pass	pass	pass	pass	pass
muxandsel (3)	pass	zero	pass	pass	pass	pass
OUT 2					$16D_{01}$	
OUT 1			$8D_{00}$		$16D_{02}$	
OUTPUT LEADS 520				D_{00}	D_{01}	D_{02}

FIG. 28A

CONV_ROW CONTROL SIGNALS AND OUTPUTS DURING INVERSE OCTAVE O TRANSFORM

6	7	8	9	10	
H_{03}	G_{03}	H_{10}	G_{10}	H_{11}	
3	2	2	4	2	
1	2	3	1	1	
1	1	1	1	1	
zero	pass	zero	pass	zero	
pass	pass	zero	pass	pass	
pass	pass	pass	pass	pass	
add	add	add	add	add	
add	sub	add	sub	add	
add	add	add	add	add	
sub	add	sub	add	sub	
r	l	r	l	r	
l	r	l	r	l	
pass	pass	pass	pass	pass	
pass	pass	zero	pass	pass	
pass	pass	pass	zero	pass	
$16D_{03}$		$16D_{05}$		$8D_{07}$	
$16D_{04}$		$16D_{06}$		$8D_{10}$	
D_{03}	D_{04}	D_{05}	D_{06}	D_{07}	D_{10}

FIG. 28B

CONV_ROW CONTROL SIGNALS AND OUTPUTS DURING INVERSE OCTAVE O TRANSFORM

KEY TO Fig. 28

Fig. 28A	Fig. 28B

Fig. 28

Time	Data Values	Output of Block 926	Output of Block 928
0	H_{00}	$32cH_{00}$	$32(b-a)H_{00}$
1	G_{00}	$8D_{00} = 32\{(b-a)H_{00} + (c-d)G_{00}\}$	$32\{cH_{00} - bG_{00}\}$
2	H_{01}	$32cH_{01}$	$32\{-dH_{00} + aG_{00} + bH_{01}\}$
3	G_{01}	$16D_{02} = 32\{-dH_{00} + aG_{00} + bH_{01} + cG_{01}\}$	$32\{cH_{01} - bG_{01}\}$
4	H_{02}	$32cH_{02}$	$32\{-dH_{01} + aG_{01} + bH_{02}\}$
5	G_{02}	$16D_{04} = 32\{-dH_{01} + aG_{01} + bH_{02} + cG_{02}\}$	$32\{cH_{02} - bG_{02}\}$
6	H_{03}	$32(c+d)H_{03}$	$32\{-dH_{02} + aG_{02} + bH_{03}\}$
7	G_{03}	$16D_{06} = 32\{-dH_{02} + aG_{02} + bH_{03} + cG_{03}\}$	$32\{(c+d)H_{03} - (a+b)G_{03}\}$
8	H_{10}	$32cH_{10}$	$32(b-a)H_{10}$
9	G_{10}	$8D_{10} = 32\{(b-a)H_{10} + (c-d)G_{10}\}$	$32\{cH_{10} - bG_{10}\}$
10			

FIG. 29A

CONV_ROW DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

Output of Block 930	Output of Block 932
	$-32dH_{00}$
$32\{-dH_{00} + aG_{00}\}$	
$32\{cH_{00} - bG_{00} + aH_{01}\}$	$32\{-dH_{01}\}$
$32\{-dH_{01} + aG_{01}\}$	$16D_{01} = 32\{cH_{00} - bG_{00} + aH_{01} + dG_{01}\}$
$32\{cH_{01} - bG_{01} + aH_{02}\}$	$32\{-dH_{02}\}$
$32\{-dH_{02} + aG_{02}\}$	$16D_{03} = 32\{cH_{01} - bG_{01} + aH_{02} + dG_{02}\}$
$32\{cH_{02} - bG_{02} + aH_{03}\}$	
	$16D_{05} = 32\{cH_{02} - bG_{02} + aH_{03} + dG_{03}\}$
$32\{(c+d)H_{03} - (a+b)G_{03}\}$	$32\{-dH_{10}\}$
$32\{-dH_{10} + aG_{10}\}$	$8D_{07} = 32\{(c+d)H_{03} - (a+b)G_{03}\}$

KEY TO FIG. 29

FIG. 29A	FIG. 29B

FIG. 29

FIG. 29B

CONV_ROW DATA FLOW FOR THE INVERSE OCTAVE 0 TRANSFORM

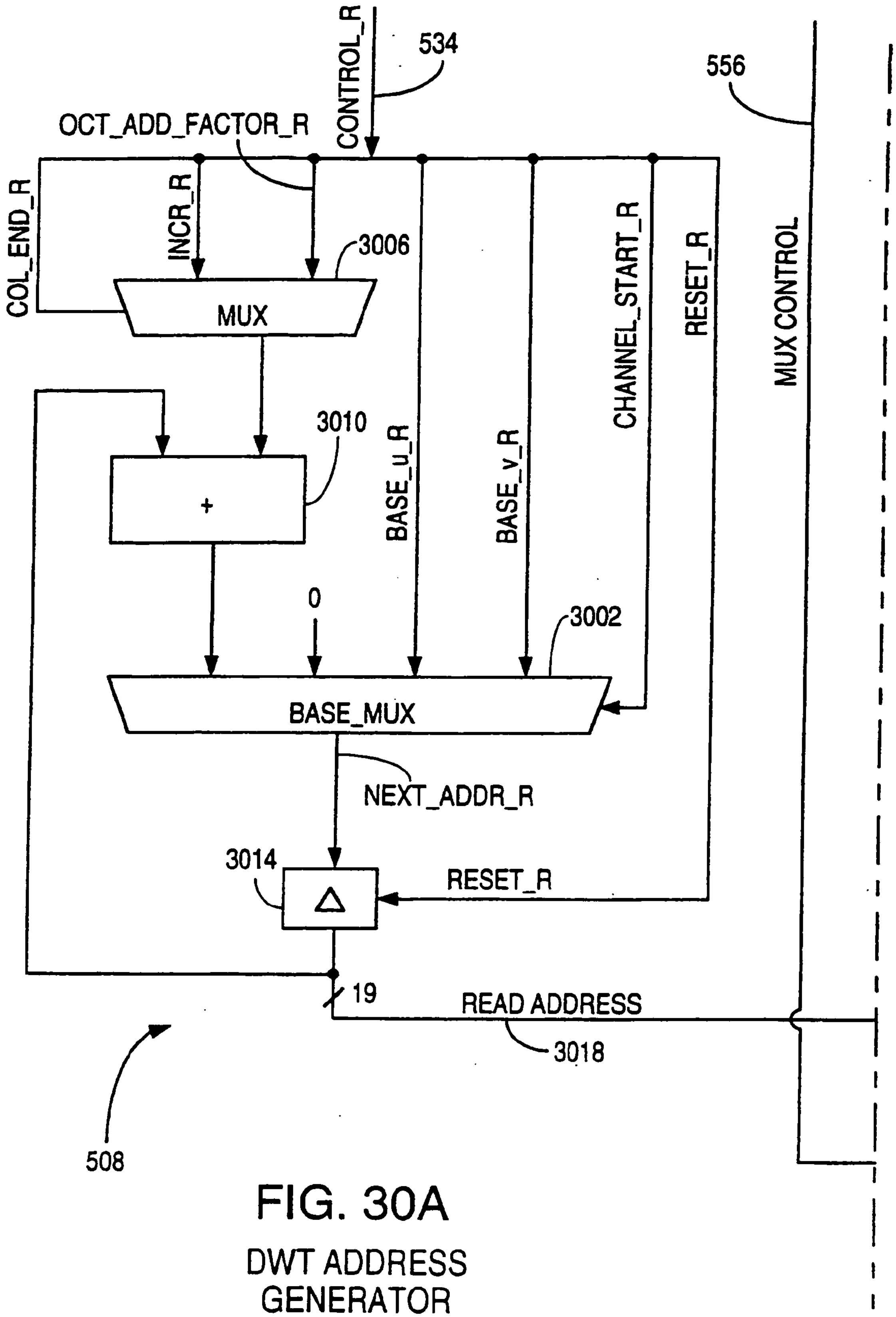


FIG. 30A

DWT ADDRESS GENERATOR

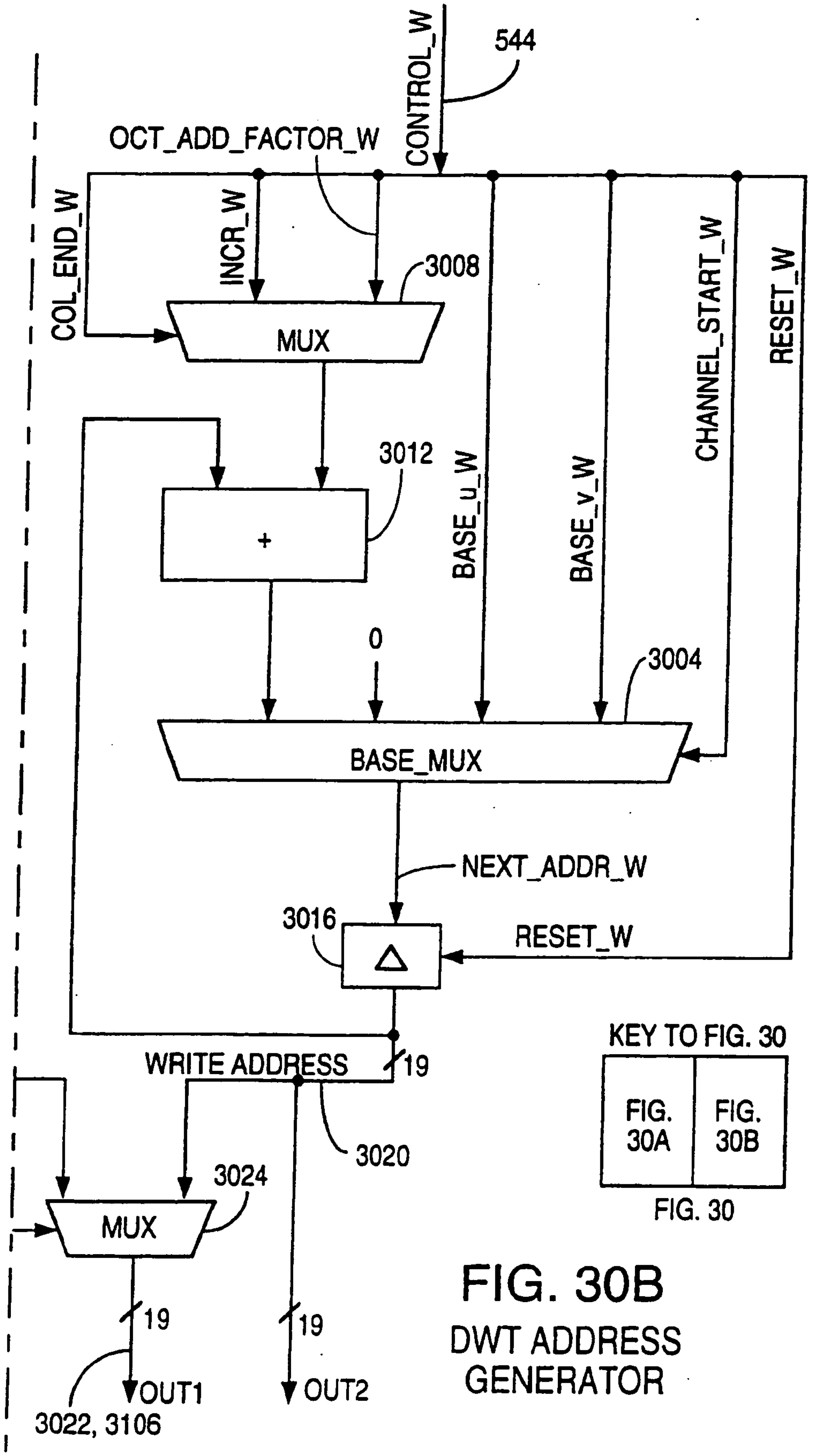


FIG. 30

FIG. 30B
DWT ADDRESS
GENERATOR

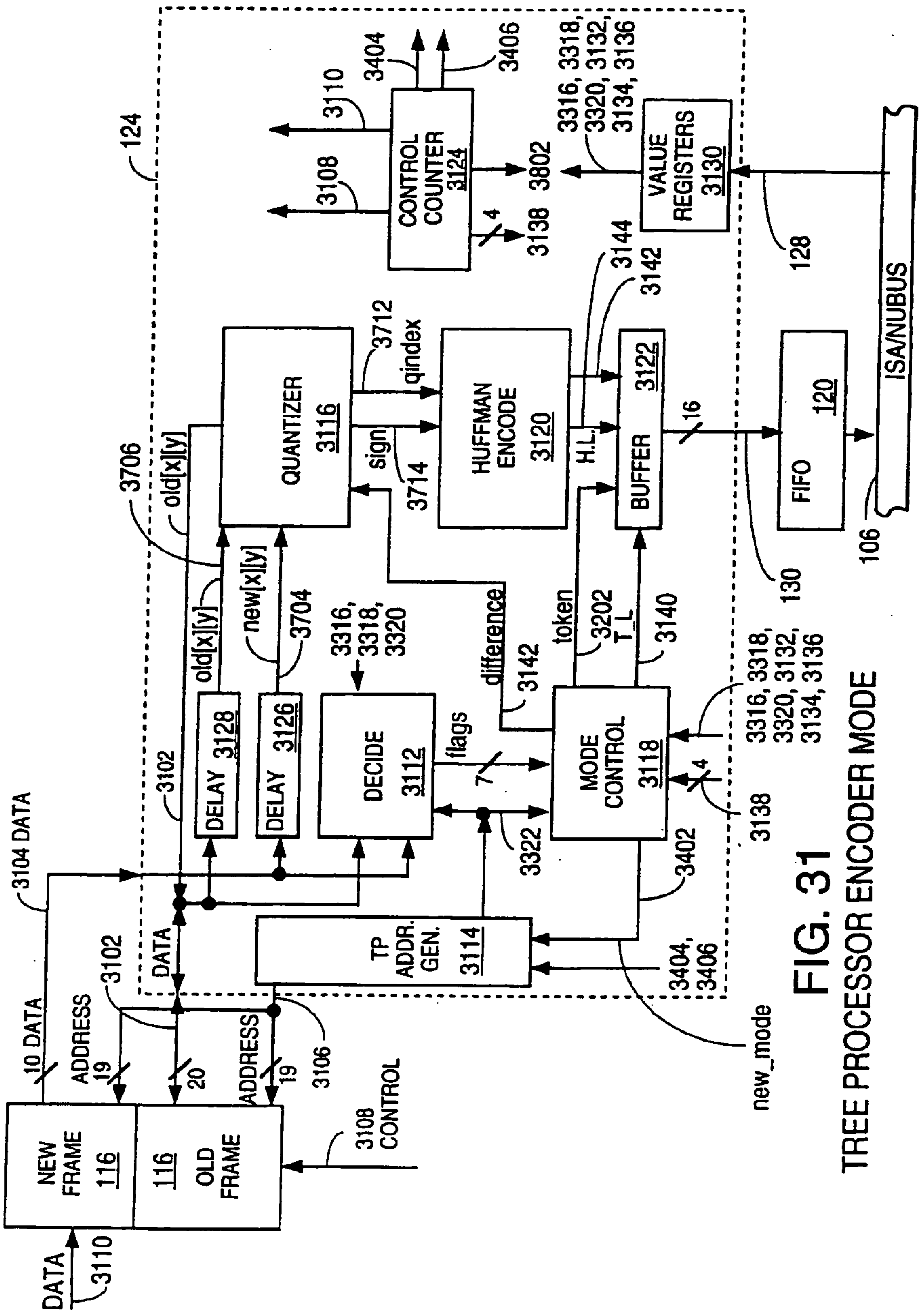


FIG. 31

TREE PROCESSOR ENCODER MODE

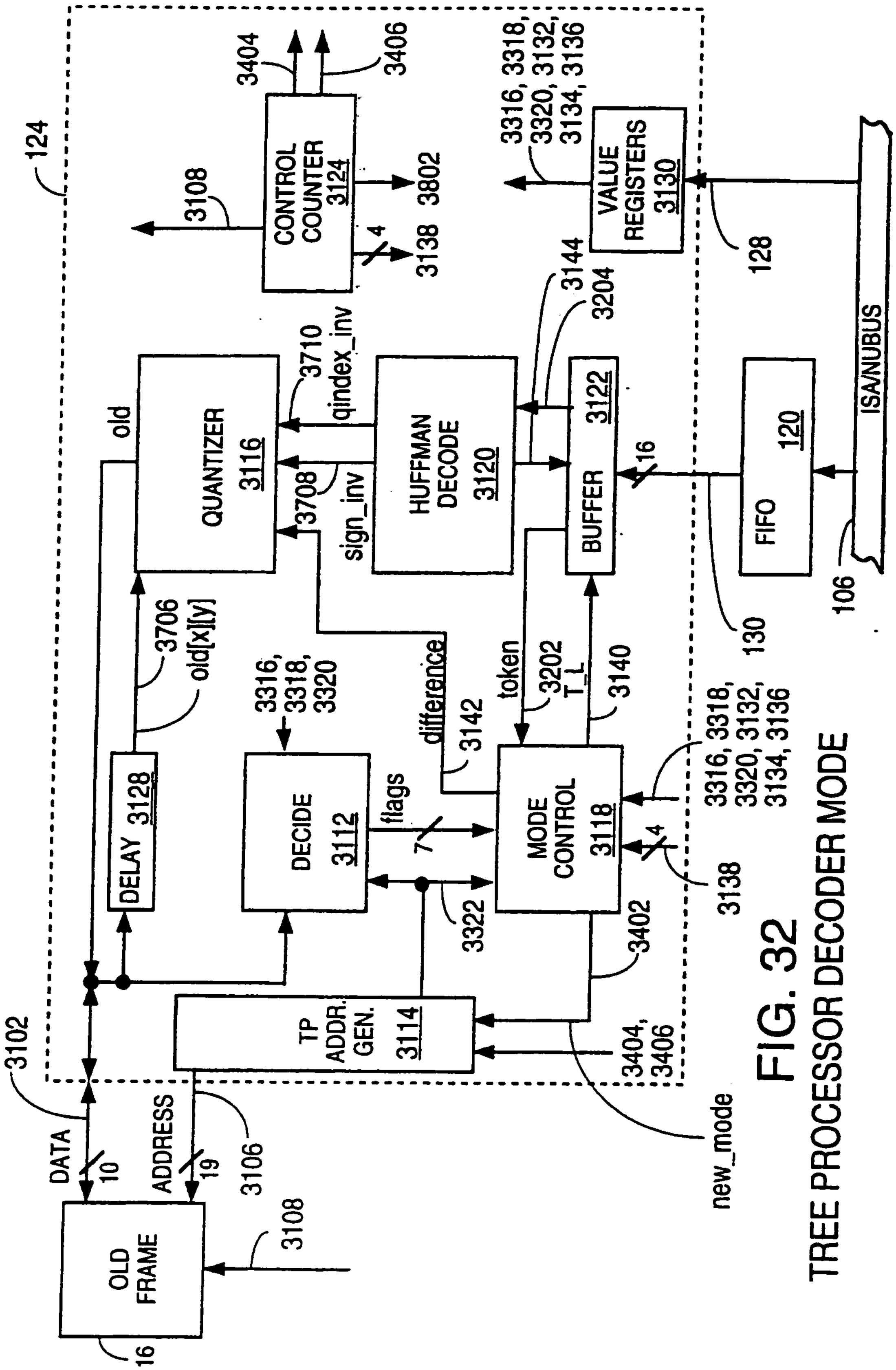


FIG. 32
TREE PROCESSOR DECODER MODE

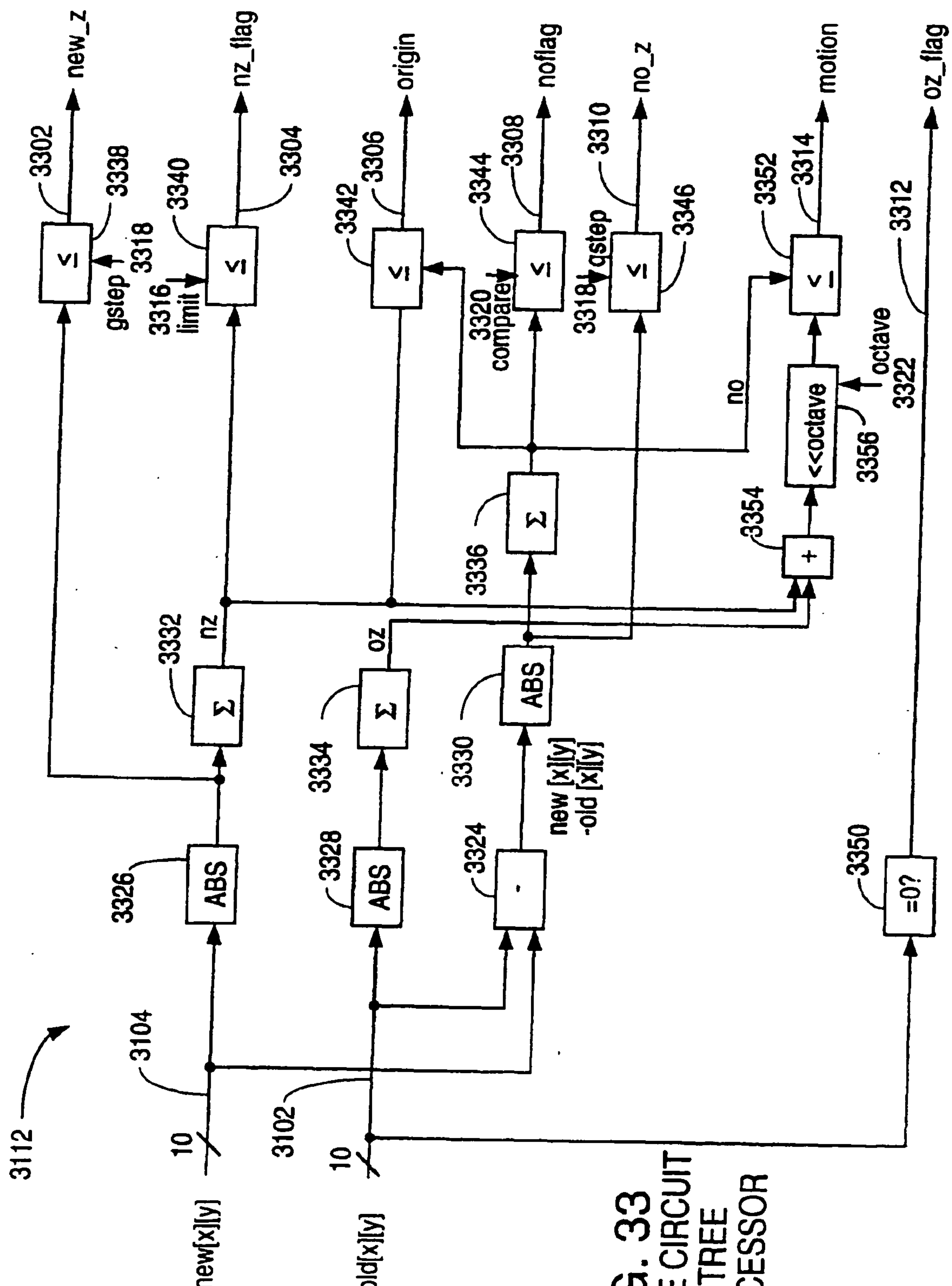


FIG. 33
DECIDE CIRCUIT
OF TREE
PROCESSOR

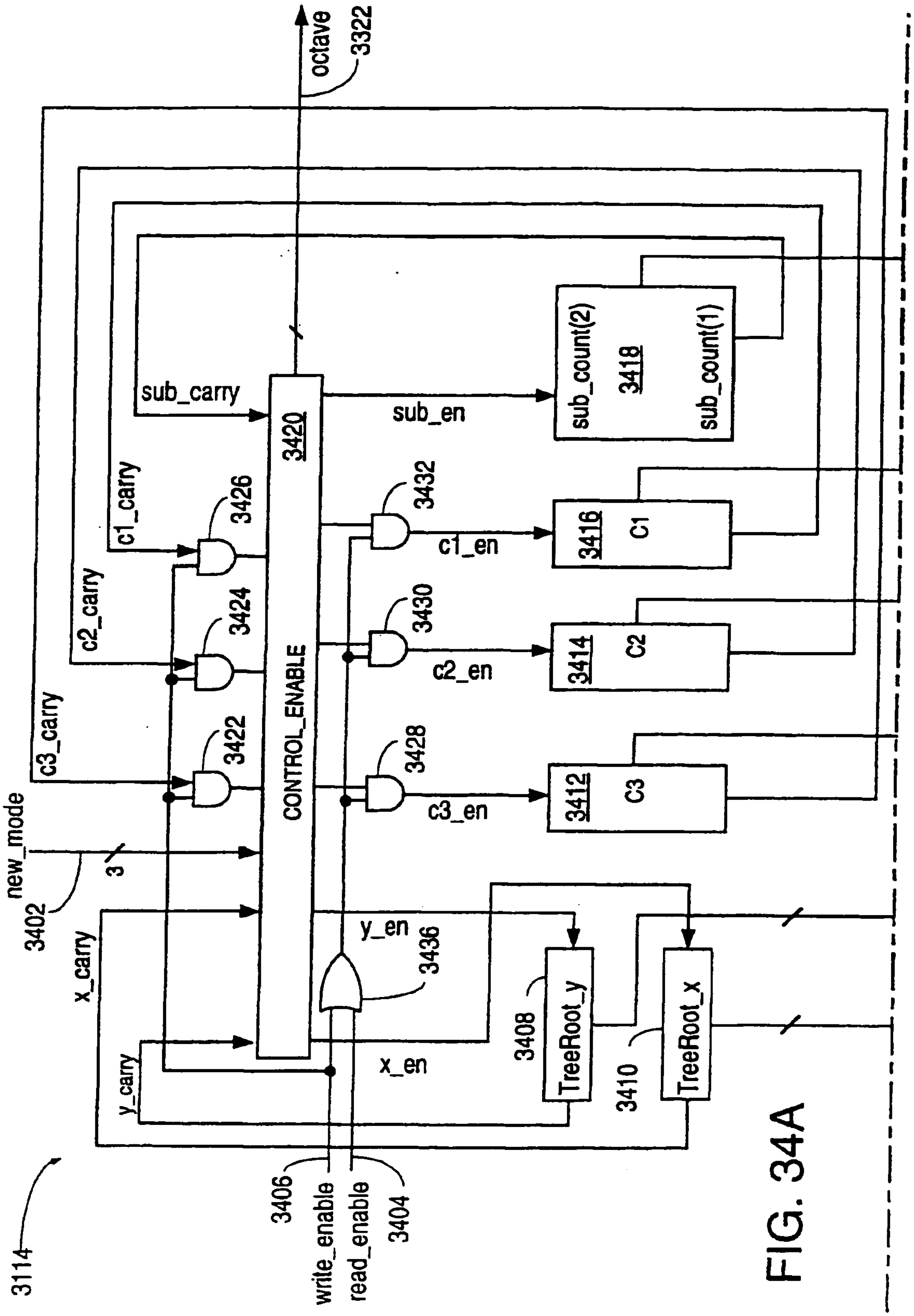

octave
3322
sub_carry
sub_en
sub_count(2)
3418
sub_count(1)
3420
CONTROL_ENABLE
c1_carry
3426
c2_carry
3424
c3_carry
3422
3432
c1_en
3416
C1
3430
c2_en
3414
C2
3428
c3_en
3412
C3
new_mode
3402
3
x_carry
y_en
3436
TreeRoot_y
3408
TreeRoot_x
3410
y_carry
x_en
3406
write_enable
read_enable
3404
3114

FIG. 34A

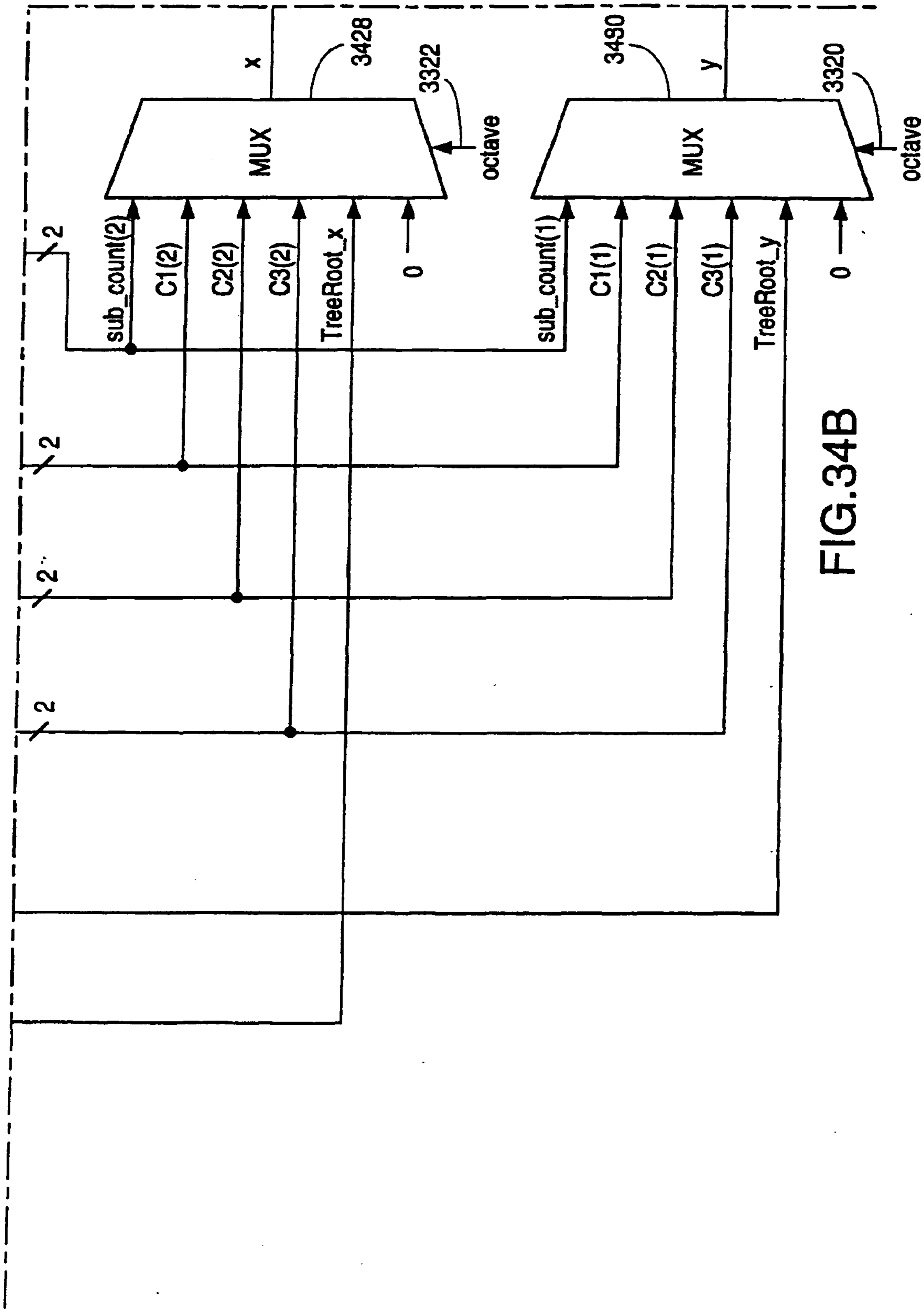

x
3428
3322
MUX
octave
3430
y
3320
MUX
octave
2
sub_count(2)
C1(2)
C2(2)
C3(2)
TreeRoot_x
0
sub_count(1)
C1(1)
C2(1)
C3(1)
TreeRoot_y
0
2
2
2

FIG.34B

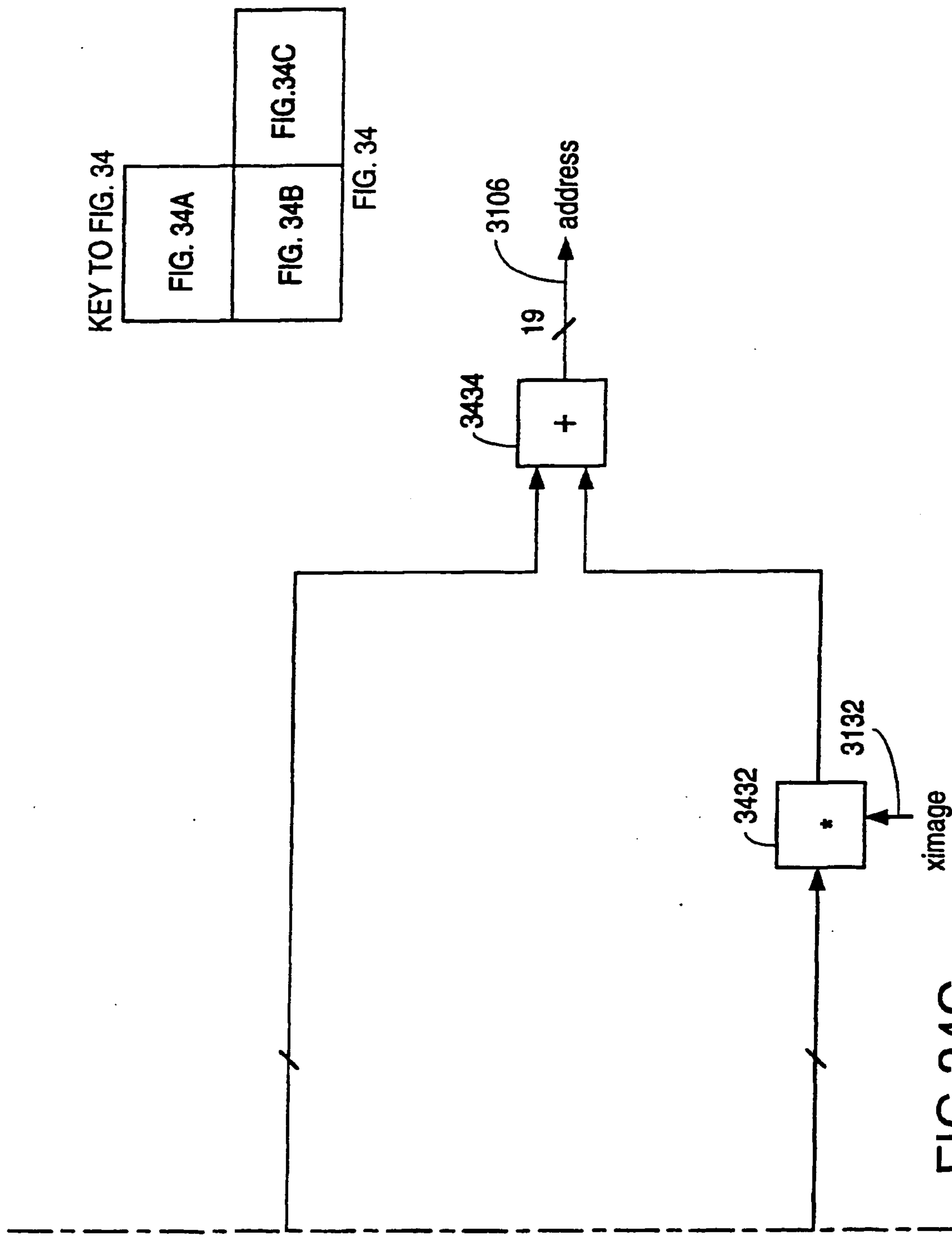

KEY TO FIG. 34
FIG. 34A
FIG. 34B
FIG.34C
FIG. 34
3106
address
19
3434
+
3432
*
3132
ximage

FIG.34C

octave	STATE OF COUNTER C3	STATE OF COUNTER C2	STATE OF COUNTER C1	RELATIVE MOVEMENT BETWEEN TREE NODES
2	(0-3)	0	0	up0
1	0	(0-3)	0	up1
0	0	0	(0-3)	zz0
0	0	1	(0-3)	zz1
0	0	2	(0-3)	zz2
0	0	3	(0-3)	zz3
1	1	(0-3)	0	down1
0	1	0	(0-3)	zz0
0	1	1	(0-3)	zz1
0	1	2	(0-3)	zz2
0	1	3	(0-3)	zz3
1	2	(0-3)	0	down2
0	2	0	(0-3)	zz0
0	2	1	(0-3)	zz1
0	2	2	(0-3)	zz2
0	2	3	(0-3)	zz3
1	3	(0-3)	0	down3
0	3	0	(0-3)	zz0
0	3	1	(0-3)	zz1
0	3	2	(0-3)	zz2
0	3	3	(0-3)	zz3
2	(0-3)	0	0	up0
1	0	(0-3)	0	up1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

FIG. 35
STATE TRANSITION DIAGRAM OF CONTROL ENABLE BLOCK OF TREE PROCESSOR ADDRESS GENERATOR

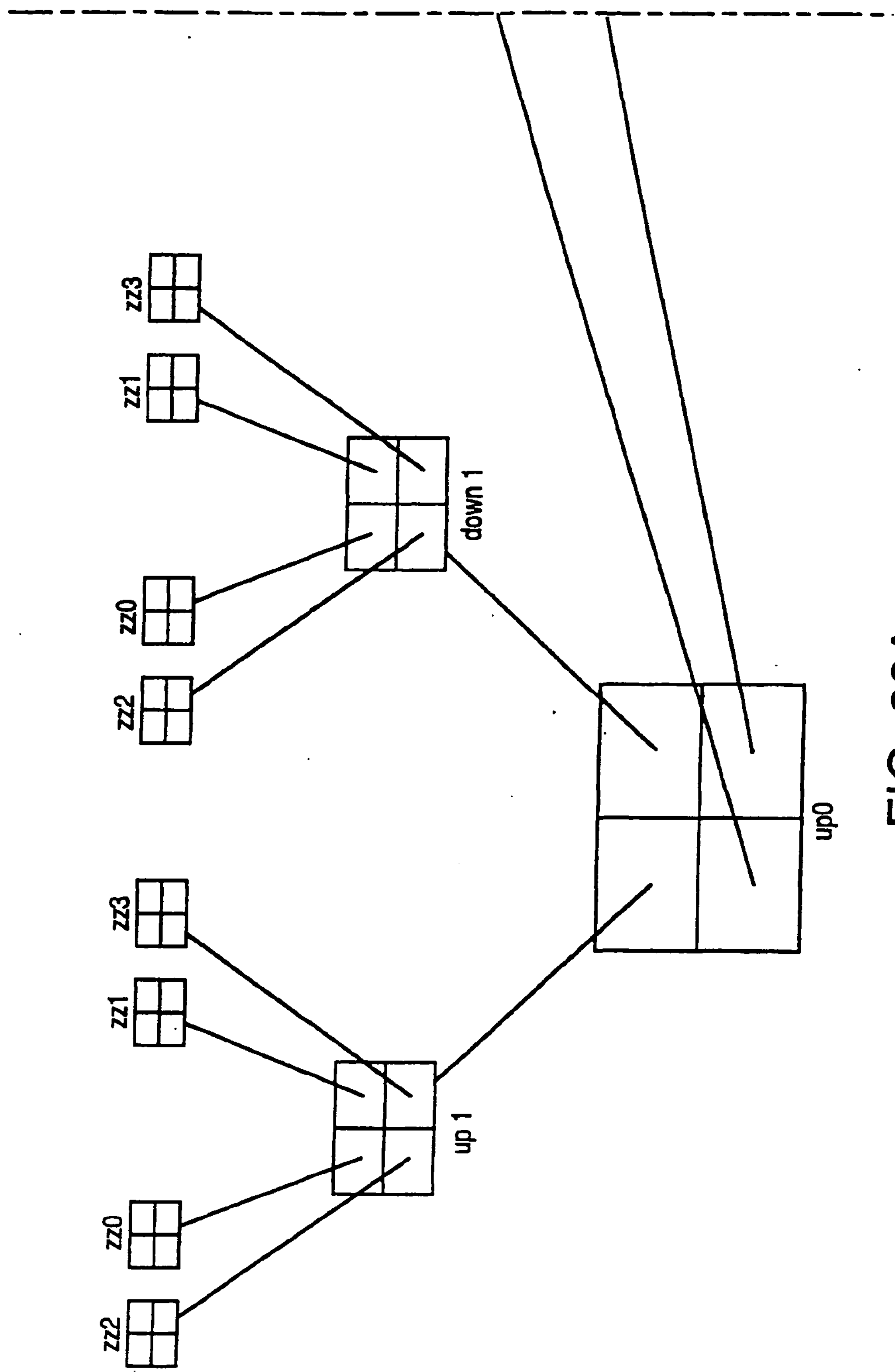


FIG. 36A
TREE DECOMPOSITION (3 OCTAVES)

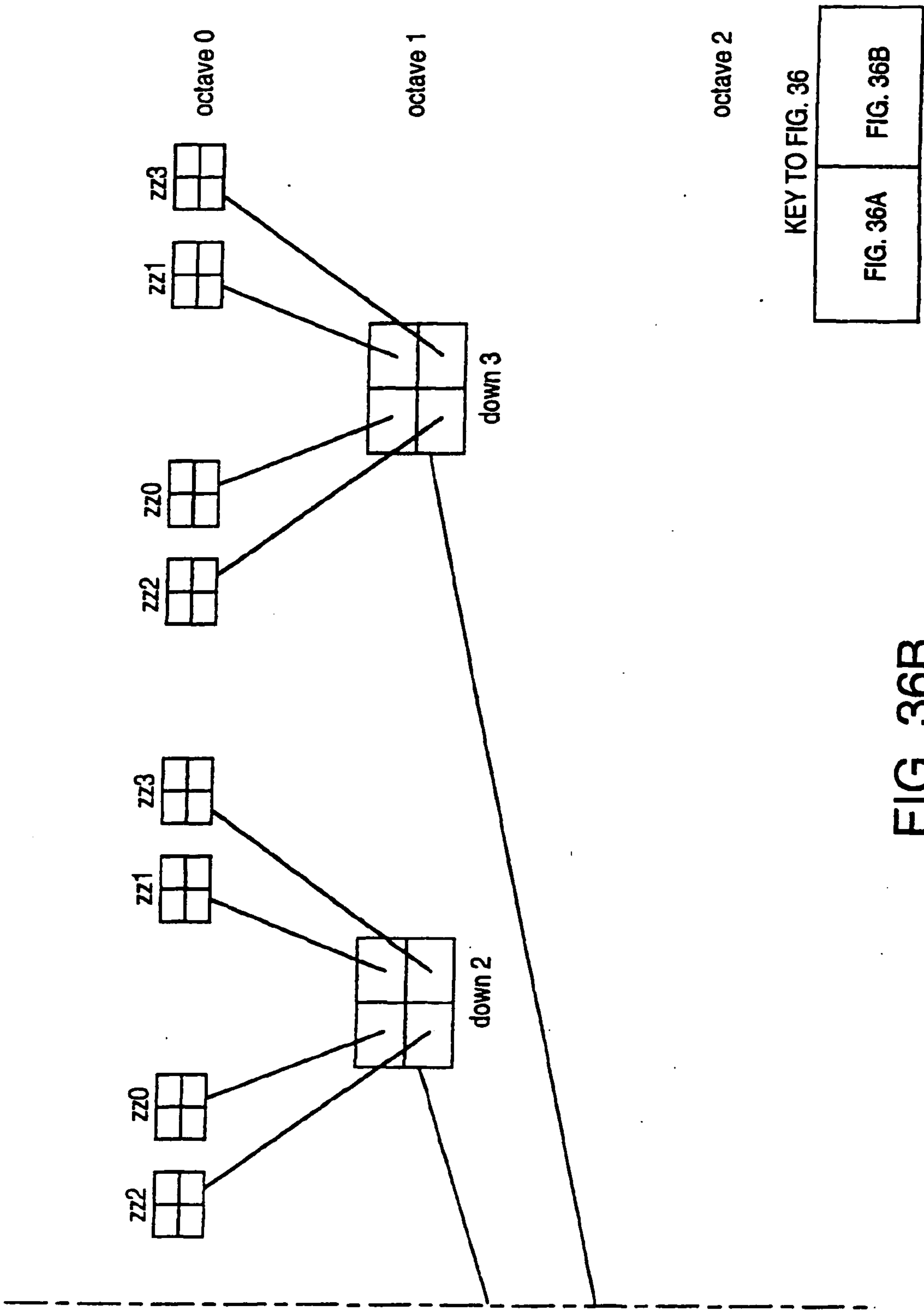


FIG. 36B
TREE DECOMPOSITION (3 OCTAVES)

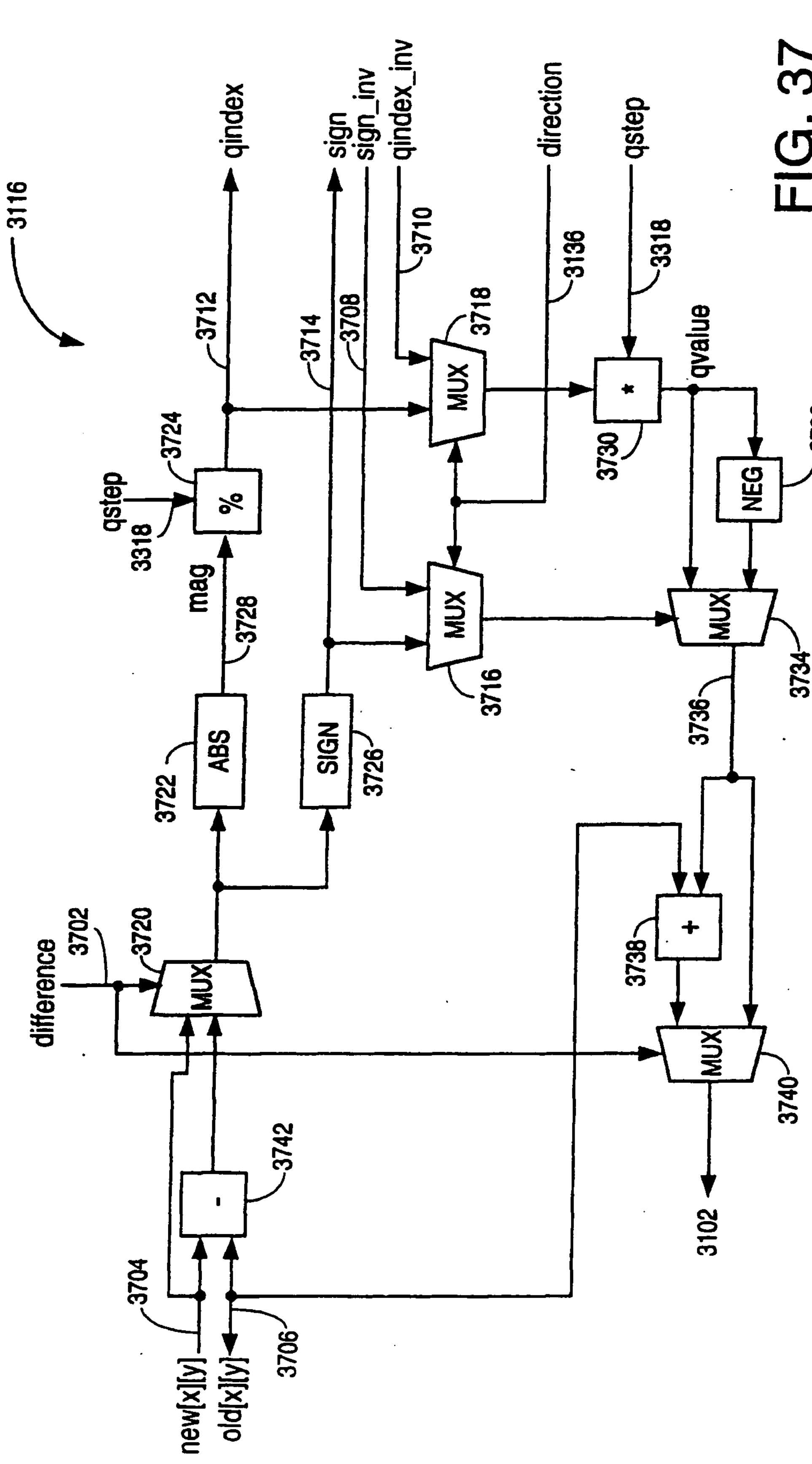


FIG. 37
QUANTIZER

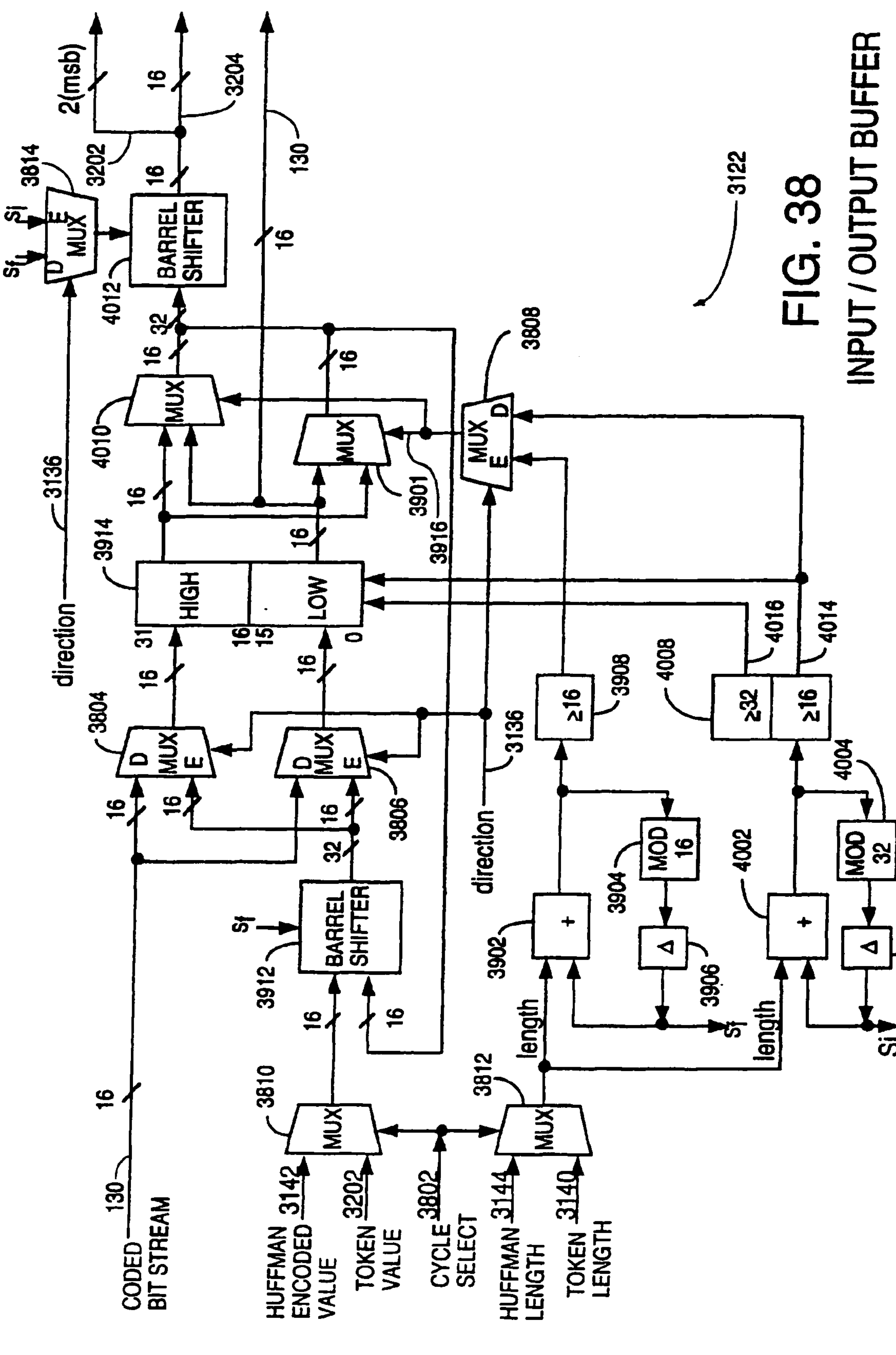


FIG. 38
INPUT / OUTPUT BUFFER

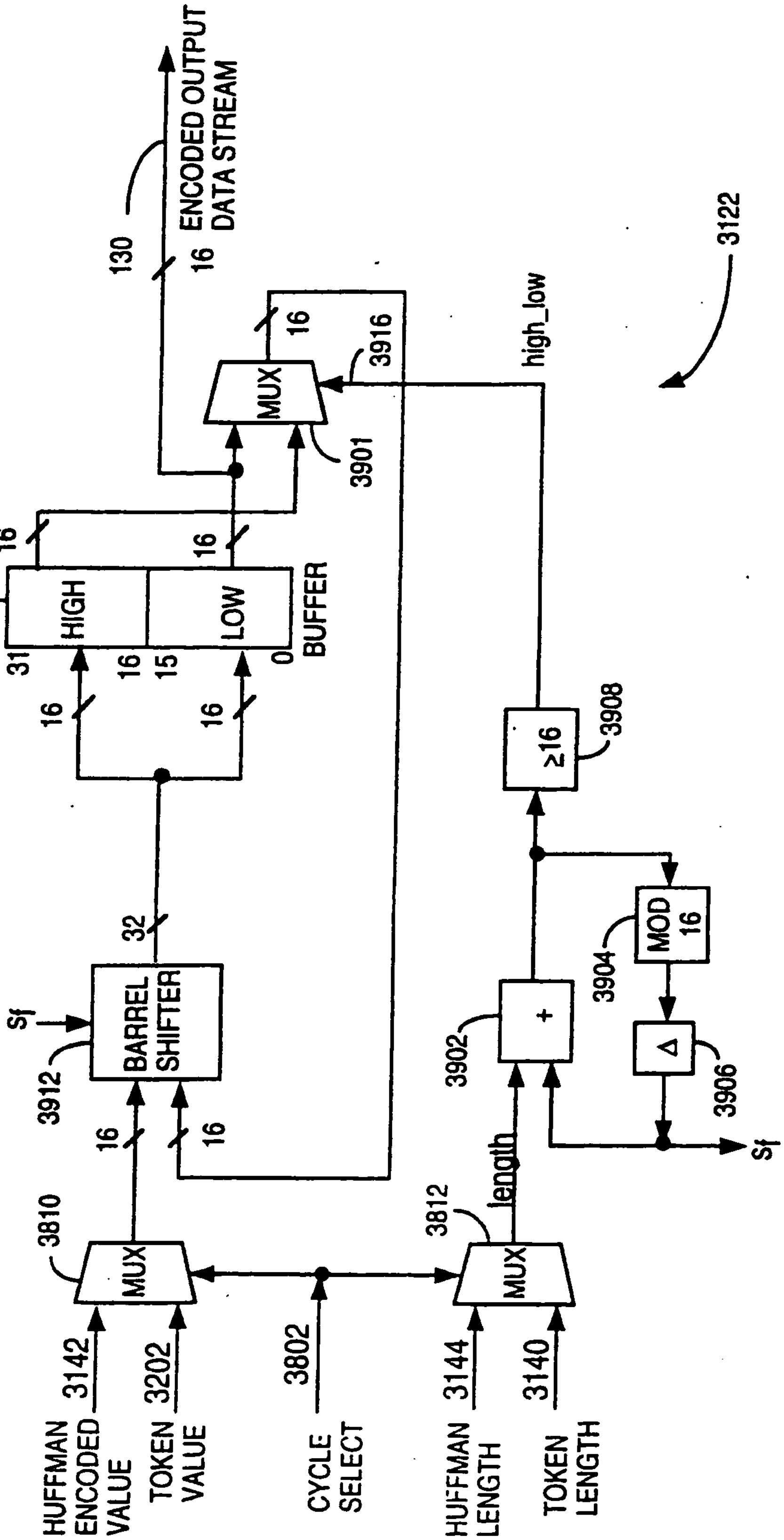


FIG. 39

BUFFER IN ENCODER MODE

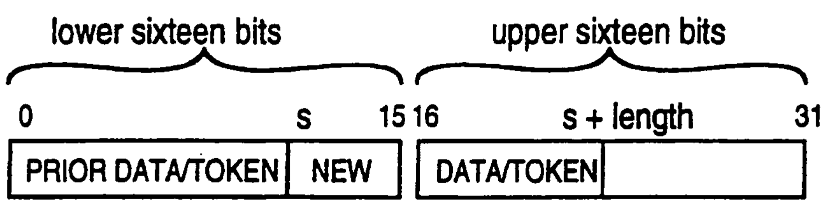


FIG. 40
OUTPUT OF BARREL SHIFTER
OF BUFFER BLOCK IN ENCODER MODE

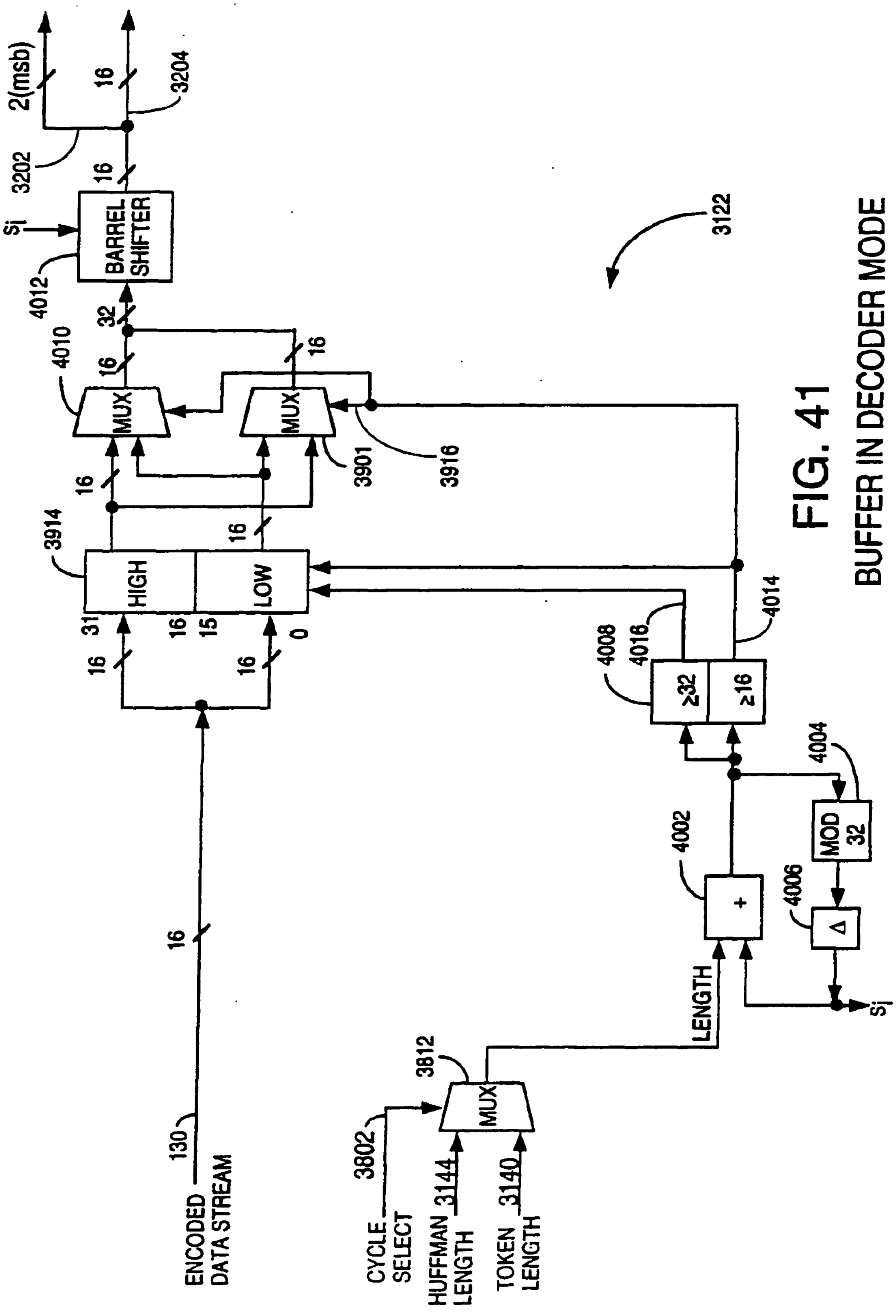


FIG. 41
BUFFER IN DECODER MODE

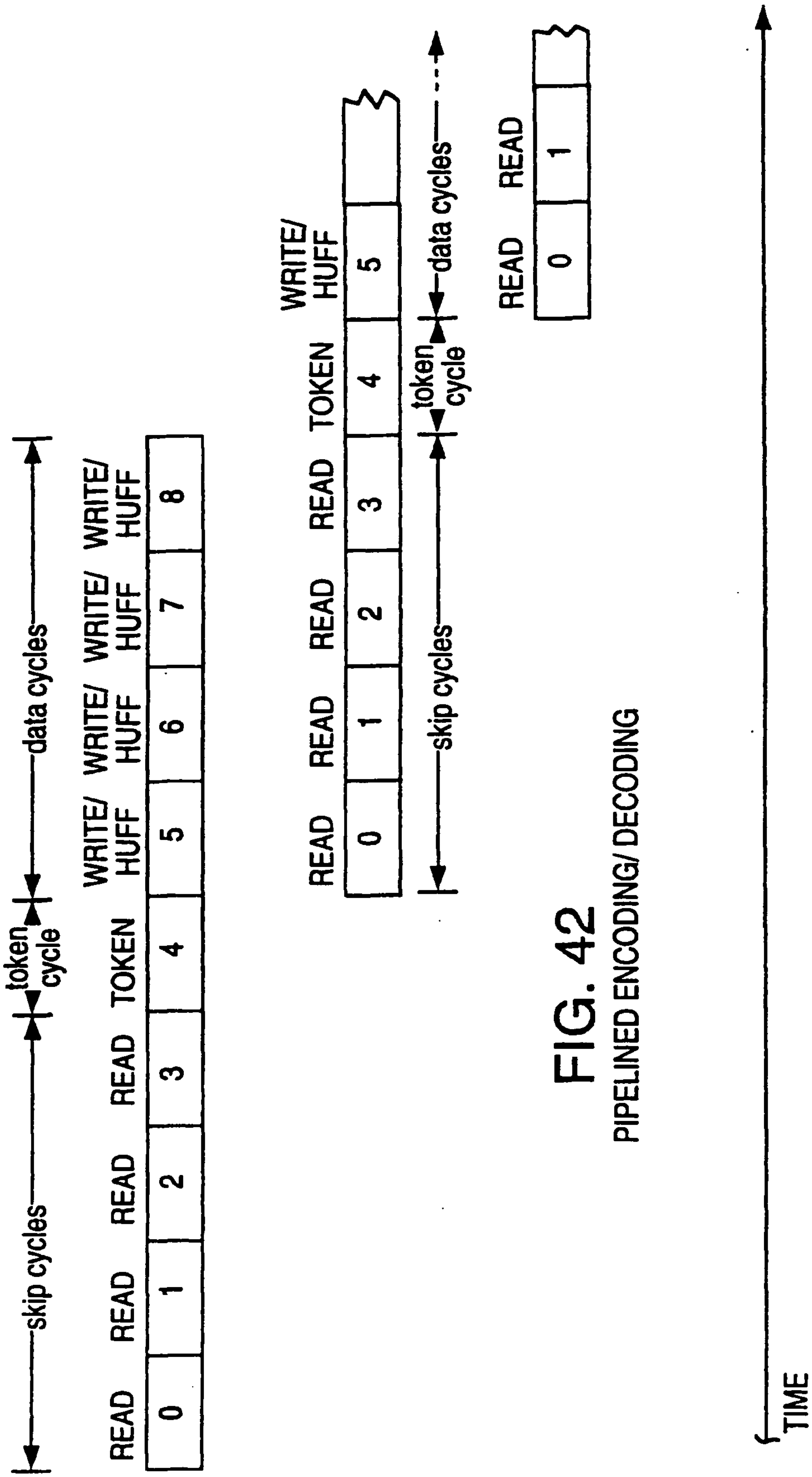


FIG. 42
PIPELINED ENCODING/ DECODING

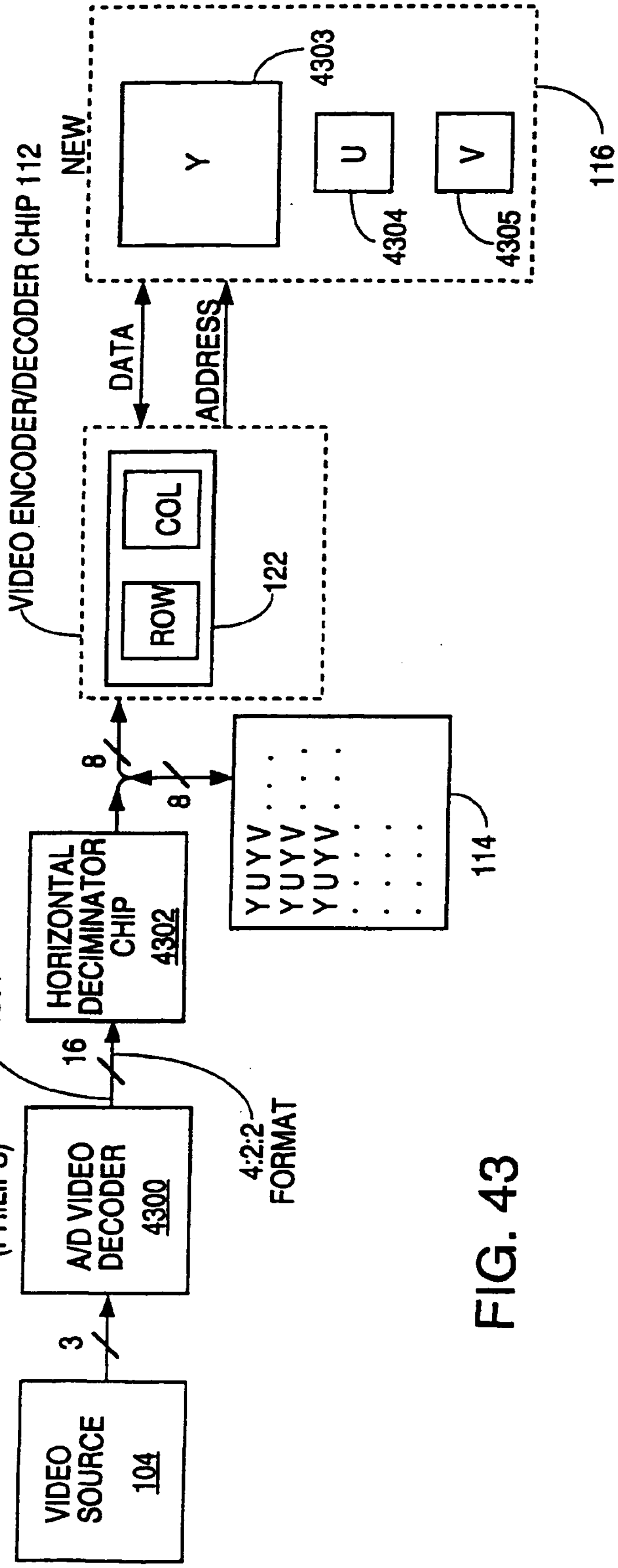
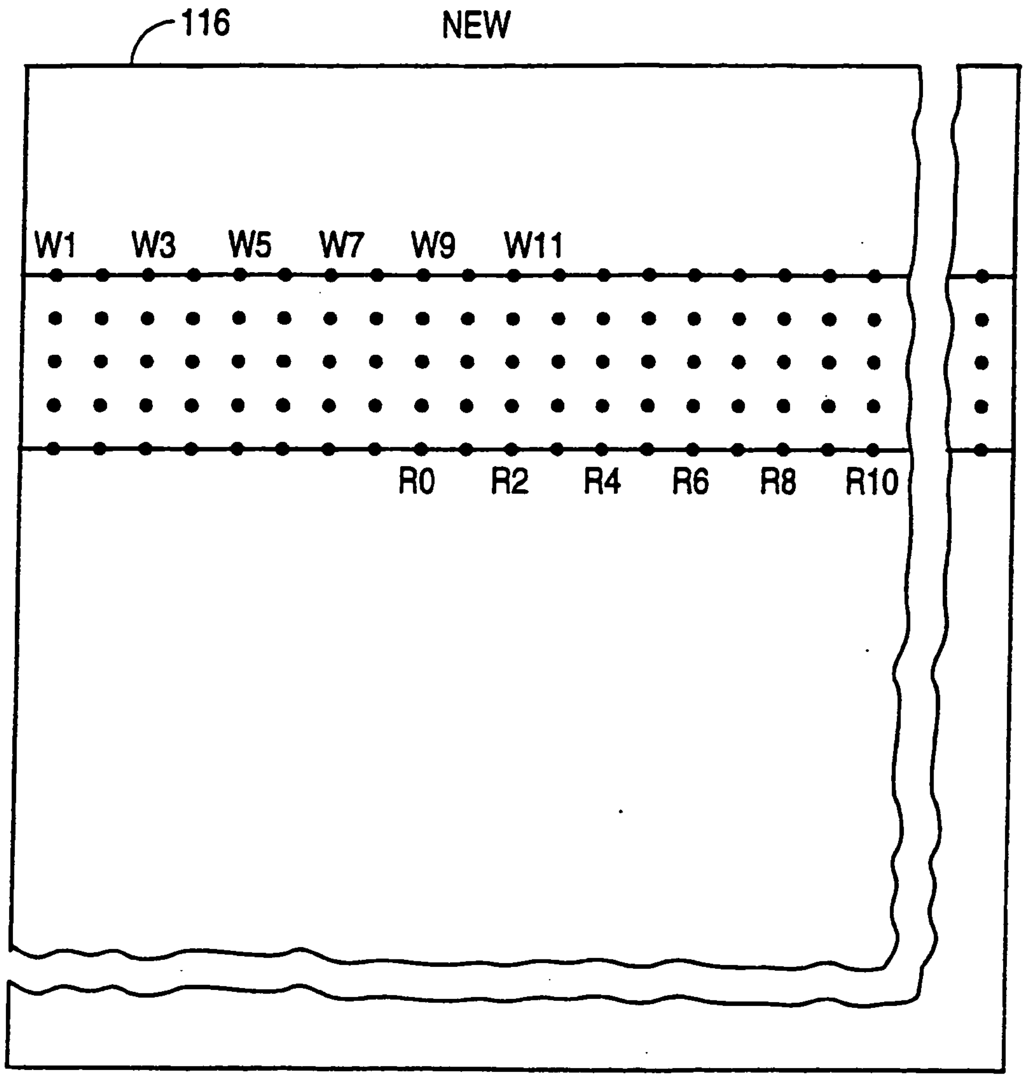
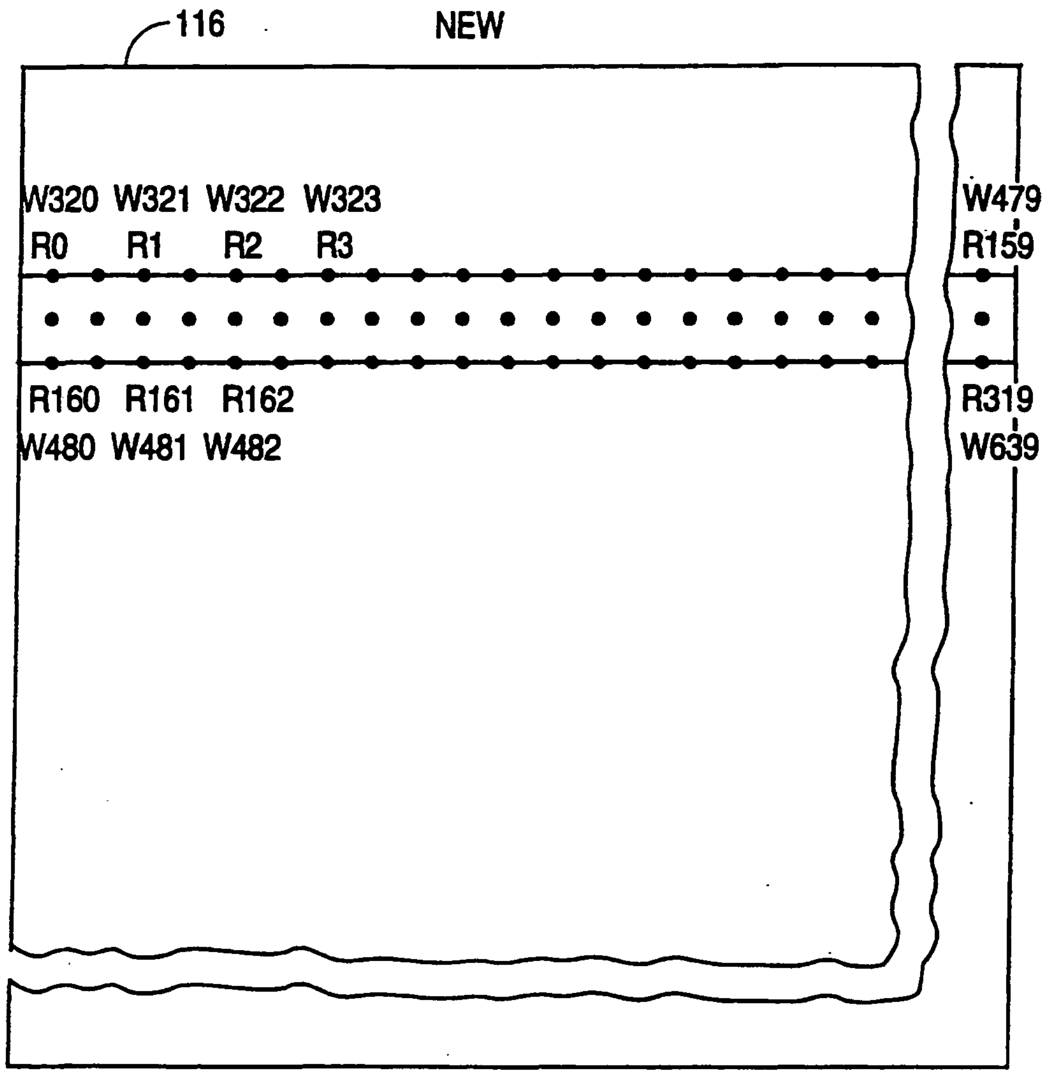

VIDEO SOURCE
104
3
(PHILIPS)
A/D VIDEO DECODER
4300
16
4301
4:2:2 FORMAT
HORIZONTAL DECIMINATOR CHIP
4302
8
8
YUYV
YUYV
YUYV
114
ROW
COL
122
VIDEO ENCODER/DECODER CHIP 112
DATA
ADDRESS
NEW
Y
4303
U
4304
V
4305
116

FIG. 43



OCTAVE 1

FIG. 44



OCTAVE 1

FIG. 45

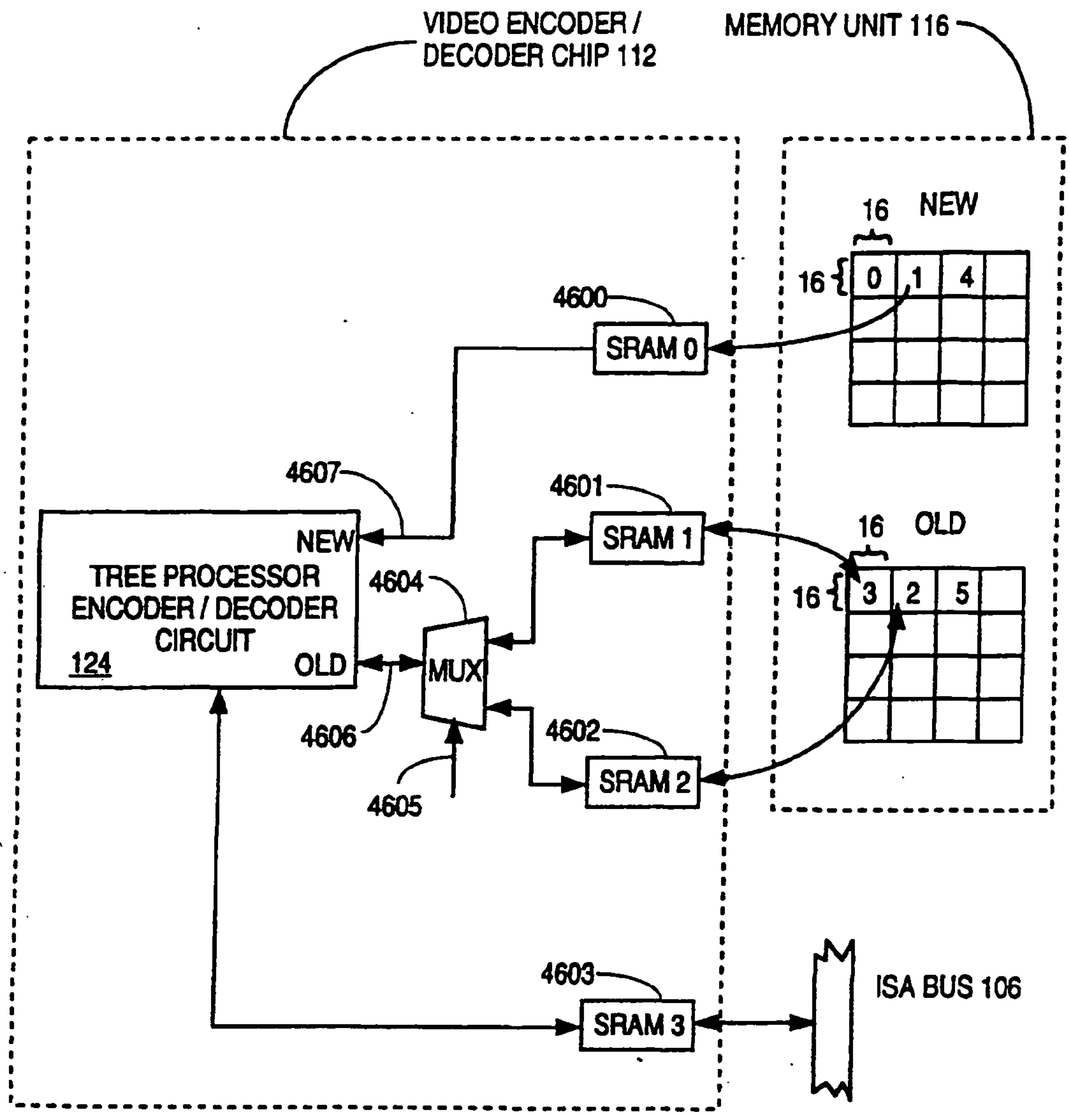

VIDEO ENCODER / DECODER CHIP 112
MEMORY UNIT 116
16
NEW
16
0
1
4
4600
SRAM 0
4607
4601
SRAM 1
16
OLD
16
3
2
5
NEW
TREE PROCESSOR ENCODER / DECODER CIRCUIT
124
OLD
4604
MUX
4606
4602
SRAM 2
4605
4603
SRAM 3
ISA BUS 106

FIG. 46

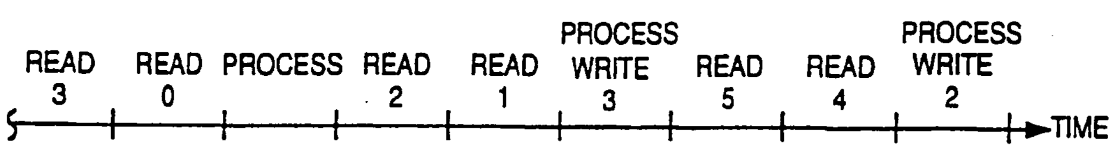

READ 3
READ 0
PROCESS
READ 2
READ 1
PROCESS WRITE 3
READ 5
READ 4
PROCESS WRITE 2
TIME

FIG. 47

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 622 741 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
30.12.1998 Bulletin 1998/53

(51) Int Cl.[6]: **G06F 15/332**, H04N 7/13, G06F 15/64

(43) Date of publication A2:
02.11.1994 Bulletin 1994/44

(21) Application number: 94302323.4

(22) Date of filing: 30.03.1994

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: 30.03.1993 US 40301
30.07.1993 US 100747
01.10.1993 US 130571

(71) Applicant: KLICS, Ltd.
Jersey JE4 8X2, Channel Islands (GB)

(72) Inventor: Knowles, Gregory P.
E-07011 Palma (ES)

(74) Representative: W.P. THOMPSON & CO.
Celcon House
289-293 High Holborn
London WC1V 7HU (GB)

(54) Device and method for data compression/decompression

(57) An apparatus produces an encoded and compressed digital data stream from an original input digital data stream using a forward discrete wavelet transform and a tree encoding method. The input digital data stream may be a stream of video image data values in digital form. The apparatus is also capable of producing a decoded and decompressed digital data stream closely resembling the originally input digital data stream from an encoded and compressed digital data stream using a corresponding tree decoding method and a corresponding inverse discrete wavelet transform. A dual convolver is disclosed which performs both boundary and nonboundary filtering for forward transform discrete wavelet processing and which also performs filtering of corresponding inverse transform discrete wavelet processes. A portion of the dual convolver is also usable to filter an incoming stream of digital video image data values before forward discrete wavelet processing. Methods and structures for generating the addresses to read/write data values from/to memory as well as for reducing the total amount of memory necessary to store data values are also disclosed.

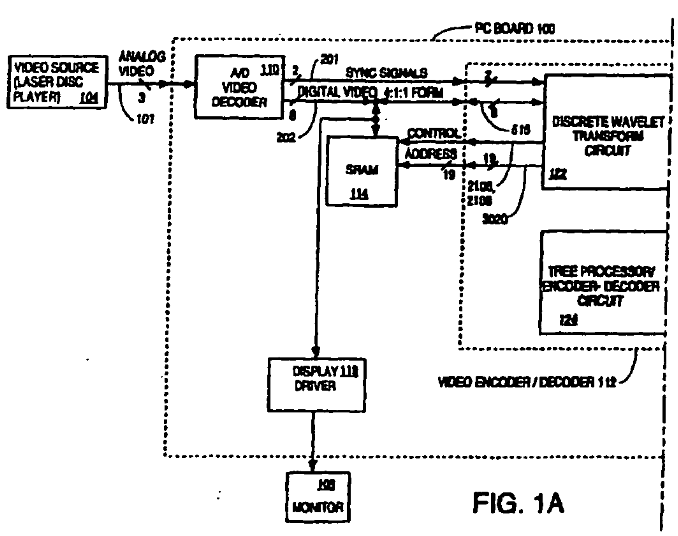


FIG. 1A

EP 0 622 741 A3

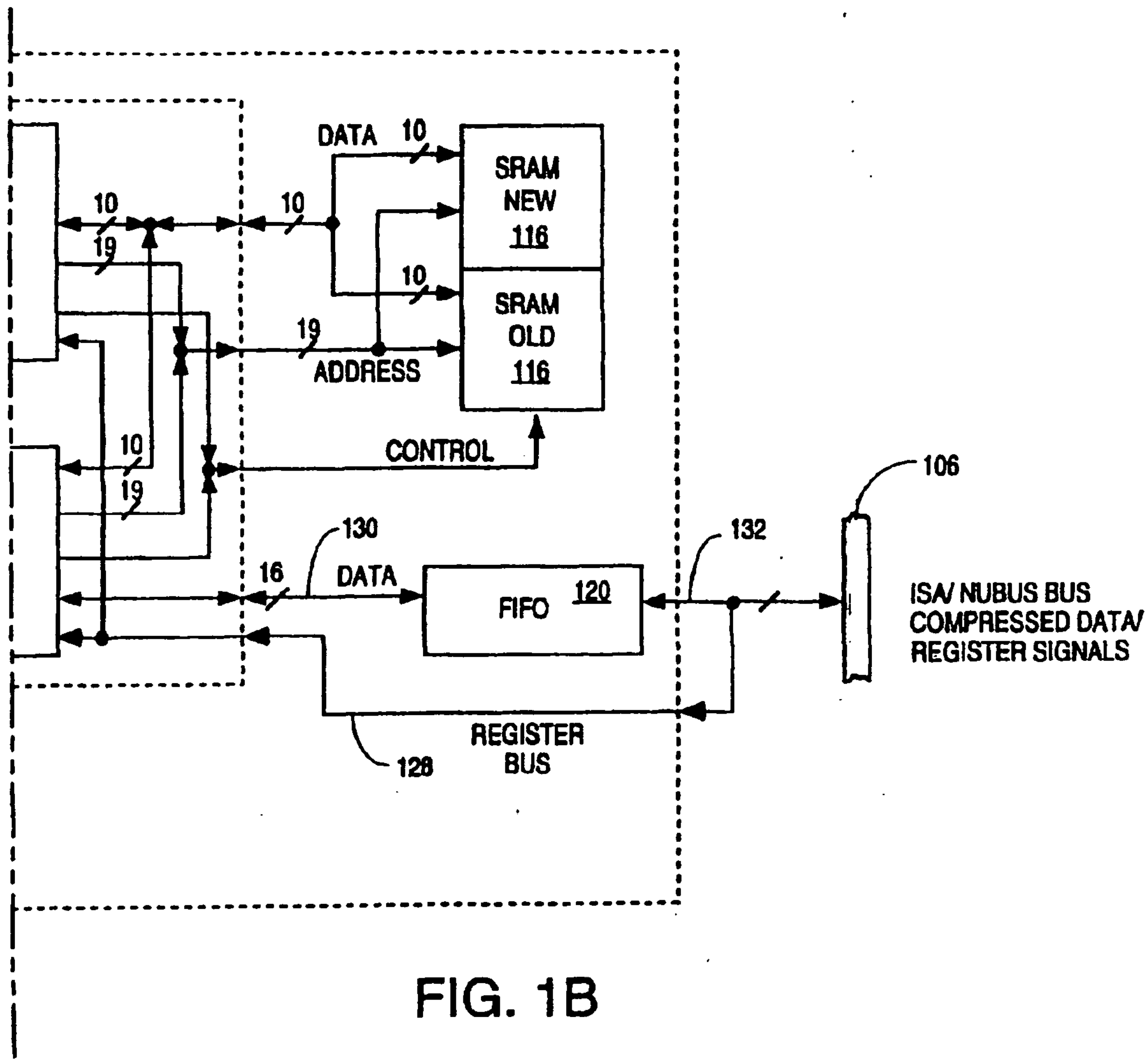

DATA
10
SRAM
NEW
116
10
10
19
10
SRAM
OLD
116
19
ADDRESS
10
CONTROL
19
106
130
132
16
DATA
FIFO
120
ISA/ NUBUS BUS
COMPRESSED DATA/
REGISTER SIGNALS
REGISTER
BUS
128

FIG. 1B

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 2323

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
X	US 4 974 187 A (LAWTON WAYNE M) 27 November 1990 * the whole document *	1-9,11, 12,24-26	G06F15/332 H04N7/13 G06F15/64
A	CHUI C K ET AL: "WAVELETS ON A BOUNDED INTERVAL" NUMERICAL METHODS IN APPROXIMATION THEORY, vol. 9, 1 December 1992, pages 53-75, XP000562102	3	
A	LEWIS A S ET AL: "VIDEO COMPRESSION USING 3D WAVELET TRANSFORMS" ELECTRONICS LETTERS, vol. 26, no. 6, 15 March 1990, pages 396-398, XP000122769		
A	LEWIS A S ET AL: "IMAGE COMPRESSION USING THE 2-D WAVELET TRANSFORM" IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 1, no. 2, 1 April 1992, pages 244-250, XP000367551		

TECHNICAL FIELDS SEARCHED (Int.Cl.5)

G06F

~~The present search report has been drawn up for all claims~~

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 July 1998	PIERFEDERICI, A

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)